

UNSUPERVISED AND SUPERVISED FUZZY
NEURAL NETWORK ARCHITECTURE,
WITH APPLICATIONS IN MACHINE
VISION FUZZY OBJECT
RECOGNITION AND
INSPECTION

By

BAOSHAN CHEN

Bachelor of Science
Zhejiang University
Hangzhou, China
1983

Master of Science
Huaqiao University
Quanzhou, China
1986

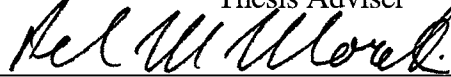
Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the degree of
DOCTOR OF PHILOSOPHY
May, 1996

UNSUPERVISED AND SUPERVISED FUZZY
NEURAL NETWORK ARCHITECTURE,
WITH APPLICATIONS IN MACHINE
VISION FUZZY OBJECT
RECOGNITION AND
INSPECTION


Thesis Approved:



Thesis Adviser









Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my major adviser Dr. Lawrence L. Hoberock for his dedication, guidance, encouragement, and endless hours of editing during the course of this work. Many thanks also go to Drs. Scott T. Acton, Glenn A. Kranzler, Eduardo A. Misawa, and Peter M. Moretti for serving on my graduate committee. Their suggestions and support were very helpful throughout this study. I would also like to extend many thanks to Dr. J. Keith Good for his help with photography. Financial support during my Ph.D. program was provided by the Oklahoma Center for Integrated Design and Manufacturing. This work is dedicated to my parents, Xiuqin Li and Shilin Chen, and my wife, Huiling Jiang, for their love, understanding, and encouragement over the years. Special thanks to my brothers for all kinds of support.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
II. AN UNSUPERVISED FUZZY NEURAL SYSTEM FOR FUZZY PATTERN CLUSTERING	5
Introduction	5
FUZART Unsupervised Clustering	7
Fuzzy Inputs, Fuzzy Outputs, and Fuzzy Weights	7
Fuzzy Subset Functions	9
Fuzzy Signal Transmission	12
F_1^a Field Activation	13
F_2^a Field Category Choice	13
Fuzzy Resonance and Reset	14
Fuzzy Weight Learning	15
Fuzzy Complement Coding	15
FUZART Decision Making Criterion	18
Simulation Results	19
Summary	29
III. A SUPERVISED FUZZY NEURAL NETWORK ARCHITECTURE FOR FUZZY CONTROL AND CLASSIFICATION	31
Introduction	31
Desired Properties of A Supervised Fuzzy Neural Network	32
FUZAMP Algorithm	33
F_1^b Field Activation	35
Fuzzy Match Tracking	36
Fuzzy Decision	37
Simulation and Results	37
Implementation of Fuzzy Rules for Fuzzy Control Systems	38
FUZAMP for Classifications.	50
Summary	59
IV. FUZZY LOGIC CONTROLLER FOR AUTOMATIC VISION PARAMETER ADJUSTMENT	60
Introduction	60
Problem Formulation	62
Fuzzy Logic Controller for Adjustment of Gain and Offset	63
Rule Base	63
Membership Functions	67
Inference and Defuzzification	70

Chapter	Page
Threshold Control	71
Experimental Study	73
Summary	76
 V. MACHINE VISION FUZZY OBJECT RECOGNITION AND DISH CLEANLINESS INSPECTION USING ARTAMP	82
Introduction	82
Fuzzy Object Recognition Using FUZAMP	85
Feature Extraction	85
Training of FUZAMP	86
Testing of FUZAMP	88
Results	89
Dish Cleanliness Inspection Using FUZAMP.	95
Summary	99
 VI. CONCLUSIONS AND RECOMMENDATIONS	101
Conclusions	101
Recommendations	102
 REFERENCES	104
 APPENDIX	107

LIST OF TABLES

Table	Page
2.1 Outputs of FUZART in Simulation 2 with $\rho = 0.6$ and $\tau \rightarrow \infty$	26
2.2 Outputs of FUZART in Simulation 2 with $\rho = 0.6$ and $\tau = 4$	27
2.3 Outputs of FUZART in Simulation 2 with $\rho = 0.6$ and $\tau = 1$	28
3.1 Available Fuzzy Rules in Simulation 4.	47
3.2 Fuzzy Rule Table Completed by FUZAMP in Simulation 4.	49
3.3 Comparison of Four Fuzzy Neural Networks	51
4.1 Linguistic Rules of the First Layer of an FLC	66
4.2 Linguistic Rules of the Second Layer of an FLC	66
5.1 Results of Application 1 Using FUZAMP and fuzzy ARTMAP	90
5.2 Training Data Set A and 25 Exemplars in Testing Set.	91
5.3 Results of Application 2 Using FUZAMP and fuzzy ARTMAP	93
5.4 Results of Application 3 for FUZAMP	94
5.5 Inspection Accuracy (%) for Dish Cleanliness Inspection Using FUZAMP and Fuzzy ARTMAP	100

LIST OF FIGURES

Figure	Page
2.1 The Architecture of FUZART	8
2.2 A Typical Fuzzy Number	8
2.3 Fuzzy Data Set for Clustering in Simulation 1	20
2.4 Hard Decision Clustering Performance of FUZART with $\rho = 0.2$	21
2.5 Hard Decision Clustering Performance of FUZART with $\rho = 0.5$	22
2.6 Hard Decision Clustering Performance of FUZART with $\rho = 0.65$	23
2.7 Hard Decision Clustering Performance of FUZART with $\rho = 0.76$	24
2.8 Fuzzy Data Set for Fuzzy Clustering in Simulation 2	25
3.1 The Architecture of FUZAMP	34
3.2 Membership Functions for "Small" and "Large"	39
3.3 Membership Functions for "Very Small" and " Very Large"	39
3.4 Membership Functions for "Shifted Small" and "Shifted Large"	39
3.5 Membership Functions for "Singleton Small" and "Singleton Large"	40
3.6 Fuzzy Outputs From the Trained FUZAMP in Simulation 1	40
3.7 Membership Functions for "Small" and "Large" in Simulation 2	42
3.8 Fuzzy Input Test Data for "Very Small" and " Very Large"	42
3.9 Fuzzy Input Test Data for "Shifted Small" and "Shifted Large"	42
3.10 Fuzzy Input Test Data for "Singleton Small" and "Singleton Large"	43
3.11 Fuzzy Outputs For the Trained FAM in Simulation 2	43
3.12 Fuzzy Trained Data in Simulation 3	45
3.13 Generalization of the Trained FUZAMP in Simulation 3	45

Figure	Page
3.14 Fuzzy Sets for Five Linguistic Values	46
3.15 Decision Boundary of a Two-Classifer for Fuzzy Inputs	53
3.16 Decision Boundary of a Three-Classifer for Fuzzy Inputs.	55
3.17 Membership Functions for "Small", "Medium", and "Large" in Simulation 7.	56
3.18 Decision Boundary of a Two-Classifer Using Only Numerical Data in Simulation 7	57
3.19 Decision Boundary of a Two-Classifer Using Both Numerical Data and Fuzzy Rules in Simulation 7	58
4.1 Configuration of a Vision-Based Robotics System	61
4.2 Gray Level Histogram of a Vision Target.	65
4.3 Fuzzy Sets of FLC	69
4.4 Minimum Gray Level Response of Camera 2	74
4.5 Maximum Gray Level Response of Camera 2.	75
4.6 Minimum Gray Level Response Without FLC	77
4.7 Maximum Gray Level Response Without FLC	78
4.8 Minimum Gray Level Response With a FLC	79
4.9 Maximum Gray Level Response With an FLC	80
5.1 Gray Level Images of Stainless Steel Soup Spoons in the FOV.	84
5.2 Binary Images of Stainless Steel Soup Spoons in the FOV.	84
5.3 Gray Level Images of Stainless Steel Tableware.	87
5.4 Binary Images of Stainless Steel Tableware.	87
5.5 Five Different Rectangularly Shaped Ceramic and Plastic Dishes.	96
5.6 Gray Level Histogram of A Dish	97

CHAPTER I

INTRODUCTION

Machine vision provides an effective method for intelligent decisions in tasks in many automation systems. Nevertheless there exist many types of uncertainty such as noise of various sorts, vagueness of representation, and non-random uncertainties in features extracted from camera data in machine vision applications. The objective of this research is to investigate a fuzzy neural network approach integrating fuzzy sets with neural network theories to model and manage uncertainty. The idea is motivated by attempting to solve a sorting and inspection problem in large commercial dish washing operations. Current dish washing operations of loading, sorting, inspecting and unloading of 20,000 or more dishes and silverware pieces per day are performed manually in hot and humid environments. This leads to non-uniformity in inspection, high operating costs, and labor turnover. Automation techniques are sought to perform most of the tasks currently done manually. While there are numerous technical issues involved in such automation, in this research we focus only on machine vision recognition and inspection problems. In the overall automation problem, machine vision represents one of the most difficult tasks. On-line inspection decisions concerning the cleanliness of dishes using machine vision produce some unique challenges. First, the binary image and gray level image of a dish are sensitive to changes in lighting conditions, power fluctuations and camera sensitivity drift. In fact, sensor output for all cameras may include imprecision and uncertainty. Second, food particles on a dish may be imaged at varying gray levels, depending on food type, particle size and shape, and particle location and orientation. Shadows caused by dish geometry may exhibit the same gray level as food particles, causing confusion in the

inspection process. Third, reliable sorting and inspection procedures may take excessive time for sufficient numbers of dishes to be simultaneously processed while in the camera Field of View (FOV). Finally, descriptions of "clean", "dirty", and "more or less clean" are ill-defined in a mathematical sense.

Compared with conventional techniques, fuzzy logic appears to provide a more powerful tool to deal with uncertainty and imprecision. Fuzzy logic was first introduced by Zadeh in 1965 as a means of representing and manipulating uncertain, imprecise, or "fuzzy" information. Zadeh's extension of set theory provided a mechanism for representing linguistic information, such as "good", "few", and "sometimes". Unlike traditional set theory, which describe "crisp" events, which are events that either do, or do not, occur, fuzzy set theory provides a mechanism to measure "the degree to which an event occurs". Recently, fuzzy logic has been applied to many areas, including image processing (Kim and Mitza, 1993), pattern recognition (Pedrycz, 1991), decision making (Rousselot and Balmat, et. al., 1993) and process control (Chiu and Chand, 1993). In Japan, approximately 1,000 commercial and industrial fuzzy systems have been successfully developed. Interest in the U.S.A and other countries has also been growing, as indicated in the past three years by the new International IEEE Conferences on Fuzzy Systems. The most conspicuous display of fuzzy logic applications occurred in Japan in the Sendai Subway Automatic Train Operation Controller, which captured the attention of control engineers world wide (Yasunobu and Miyamoto, 1985). Fuzzy logic has also been successfully implemented in home electric appliances such as refrigerators, washers, vacuum cleaners, air-conditioners and rice cookers. Fuzzy logic can also be applied in pattern recognition problems to deal with uncertainty.

A second powerful technology, neural networks, also appear to offer advantages for machine vision applications. Neural networks have a massively parallel structure, composed of many processing elements connected to each other through multiplicative weighting parameters, called "weights" (Carpenter and Grossberg, et. al., 1987), and are

modeled on biological neural systems. They have faster response and higher performance than sequential digital computer techniques in emulating the capabilities of the human brain. Neural networks are designed to learn, based on the various input patterns they receive. There are two broad classes of learning for all neural networks, namely supervised learning and unsupervised learning. In supervised learning, a set of training input patterns is presented to the neural network, and as each input pattern is presented, a "teacher" provides the network with the desired output for that input pattern. The neural network must then modify certain of its weights so that it can produce the desired output. After undergoing a sufficient number of such learning cycles with an appropriate array of input patterns, the network should be able to generalize, producing the desired output even for inputs that were not part of the training set. In contrast, for unsupervised learning, there is no teacher present to help distinguish right from wrong. Instead, the unsupervised neural network must distinguish on its own among different input patterns and how they should be interpreted.

In this research we combine the features of fuzzy systems (with ability to process fuzzy information) and the features of neural networks (with learning ability and high speed parallel structure) to form new fuzzy neural networks to handle uncertainty, with both unsupervised and supervised learning. Recently there has been considerable interest in the combination of neural networks and fuzzy logic systems. Most of the published work has concentrated on methods for implementing fuzzy rules in a neural network framework and for implementation in control system applications. For supervised learning problems, some approaches have targeted specific applications, such as processing linguistic information with fuzzy inputs, fuzzy outputs, and fuzzy weights. For unsupervised learning problems, almost all existing learning methods deal only with non-fuzzy examples and numerical experiments. In the work proposed herein, we explore approaches to both unsupervised and supervised learning in fuzzy neural systems which can be activated by linguistic training signals. We first propose an unsupervised fuzzy neural network for self-

organizing fuzzy clustering to find natural groupings among fuzzy input exemplars. We next attack fuzzy supervised learning problems by training the proposed network with desired fuzzy input-output linguistic information. We then apply the new fuzzy supervised neural network to machine vision for recognition of "fuzzy" objects. We also correlate human evaluations with machine evaluations to evaluate the cleanliness of dishes using the new fuzzy supervised neural network.

In the following chapters, we first describe a new unsupervised fuzzy neural system called FUZART for fuzzy pattern clustering. Following this, we extend FUZART to form a new supervised fuzzy neural network architecture called FUZAMP for fuzzy control and classification. Before we employ the new fuzzy neural networks in machine vision applications, we describe a multi-layer, multi-input, multi-output fuzzy logic controller for automatic adjustment of the vision parameters "gain" and "offset" to compensate for power fluctuations, changes in ambient light, and camera sensitivity drift in our machine vision system. This was necessary to insure reliability and accuracy in automated recognition and inspection. Chapter V applies both FUZAMP and the so-called fuzzy ARTMAP (Carpenter and Grossberg, et al., 1991) neural networks in real machine vision applications to recognize highly specular silverware and to inspect the cleanliness of dishes using the same training and testing sets and the same features for each network. Results and conclusions are given in Chapter VI.

CHAPTER II

AN UNSUPERVISED FUZZY NEURAL SYSTEM FOR FUZZY PATTERN CLUSTERING

Introduction

Fuzzy neural networks integrate fuzzy sets with neural network theories. This integration provides a fuzzy logic system with the powerful computation and learning capabilities of neural networks, and provides neural networks with "human-like" reasoning capabilities of fuzzy logic systems. Unsupervised neural networks can be employed to cluster or categorize patterns into natural groups such that patterns within a cluster are, in some sense, more similar to each other than patterns belonging to other clusters. Almost all existing learning methods for unsupervised neural networks are designed to process only numerical or crisp (i.e., non-fuzzy) data, although some unsupervised neural networks using fuzzy set-theoretic concepts have been recently proposed (Mitra and Pal, 1994; Pham et. al., 1994; Simpson, 1993). Nevertheless, these approaches employ only membership values as input and/or output, while network weighting parameters are maintained as crisp numbers. Simpson (1993) proposed an unsupervised neural network called a fuzzy min-max clustering neural network to cluster numerical data using an expansion-contraction learning algorithm. Pham and Bayro-Corrochano (1994) used the Kohonen feature-mapping algorithm and the back-propagation algorithm to cluster numerical data. The outputs of their neural networks are membership values. Mitra and Pal (1994) proposed a self-organizing neural network based on Kohonen's model to process linguistic information. The inputs and outputs of their neural network are membership values of a linguistic variable. In many pattern recognition and

decision making tasks, the available patterns or data are often ill-defined or uncertain, such that information cannot be represented directly as numerical data. In such cases, it is convenient to employ fuzzy patterns to describe input feature information. Therefore, it is desirable for an unsupervised neural network to be capable of processing fuzzy data and linguistic information. In this chapter, we propose a new unsupervised fuzzy neural network called FUZART to handle fuzzy patterns, based on Adaptive Resonance Theory (ART) networks proposed by Carpenter and Grossberg (1987). FUZART extends the so-called fuzzy ART network (Carpenter and Grossberg et al., 1991b) to process fuzzy patterns by incorporating fuzzy set operations at various stages. Since crisp data can be represented by fuzzy singletons, the proposed network can also handle fuzzy data and crisp data simultaneously. FUZART stabilizes its pattern clusters using only a few training epochs (An epoch represents one presentation of a fuzzy data input set). It provides the ability to incorporate new fuzzy data and add new clusters without retraining. FUZART employs both "hard" decision and "soft" decision mechanisms to cluster unknown fuzzy input patterns. A hard decision results in either "0" or "1", that is, a fuzzy pattern is either in a class or it is not. A soft decision allows a fuzzy pattern to belong in varying degrees to multiple clusters by providing membership values that represent the degree to which a fuzzy pattern belongs to a class. In many higher-level decision making tasks, soft decisions can be extremely useful.

FUZART proposed in this chapter passes through two major stages, namely, self-organization and decision making stages. During self-organization, a set of fuzzy patterns is employed by the network to self organize the fuzzy weights using a "winner-take-all" strategy. During decision making, fuzzy patterns are presented to the stabilized network to provide either a hard decision or a soft decision regarding cluster membership.

FUZART Unsupervised Clustering

Fig. 2.1 illustrates the architecture of FUZART. FUZART autonomously classifies fuzzy input patterns and codes each class by a fuzzy template pattern formed through the FUZART learning process. As illustrated in Fig. 2.1, m designates the number of nodes in the F_1^a field; n designates the number of nodes in the F_2^a field; each node in the F_1^a field represents a fuzzy number and is designated by the index i ; each node in the F_2^a field also represents a fuzzy number and is designated by the index j . When a fuzzy input pattern appears over the F_0^a field, the F_1^a field is activated and initially sends the fuzzy pattern to each F_2^a node j through a bottom-up fuzzy weight w_j . The F_2^a field chooses a "winner-take-all" node J that receives the maximum fuzzy input. The winner J reads out its fuzzy template as a top-down fuzzy weight z_j over the F_1^a field. If z_j mismatches the fuzzy input pattern, the F_2^a field is reset to reject the winner. A new winner is then chosen to read out its fuzzy template. The reset is repeated until a new fuzzy template matches sufficiently the fuzzy input pattern. A sufficient match, called "fuzzy resonance", accepts the current winning F_2^a node to represent the fuzzy input pattern.

Fuzzy Inputs, Fuzzy Outputs, and Fuzzy Weights

A fuzzy input pattern is an m -dimensional fuzzy vector denoted by $\mathbf{p}=(p_1,p_2,\dots,p_m)$. Let $\mathbf{x}=(x_1,x_2,\dots,x_m)$ denote the m -dimensional fuzzy output vector of the field F_1^a , and $\mathbf{y}=(y_1,y_2,\dots,y_n)$ denote the n -dimensional fuzzy output vector of the field F_2^a , each node in the field F_2^a has its own bottom-up fuzzy weight $\mathbf{w}_j=(w_{1j}, w_{2j},\dots,w_{mj})$ and top-down fuzzy weight $\mathbf{z}_j=(z_{j1}, z_{j2},\dots,z_{jm})$ for $j=1,2, \dots, n$. In FUZART, w_j is the same as z_j . Each component of a fuzzy vector is a fuzzy number. In the work herein, for simplicity, we represent a fuzzy number, illustrated in Fig. 2.2, by triangular membership functions, $\mu_A(x)$ given by:

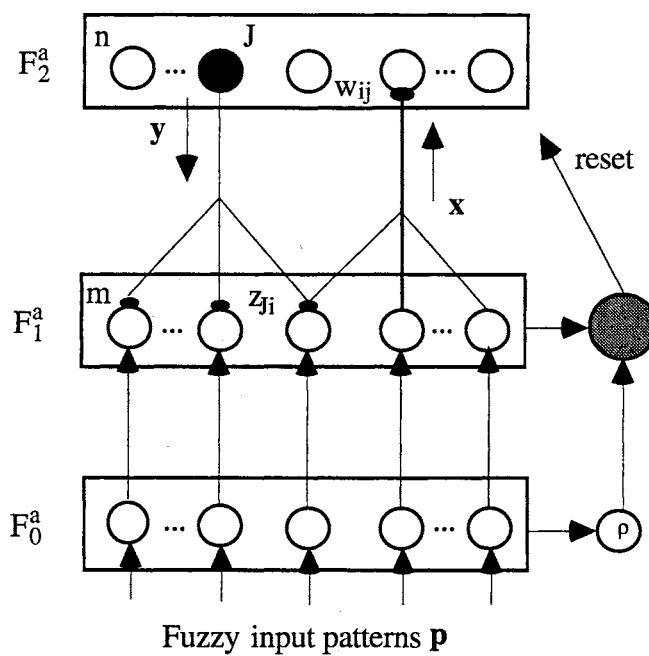


Fig. 2.1 The Architecture of FUZART
 [Adapted from ART (Carpenter, Grossberg, et al., 1991b)]

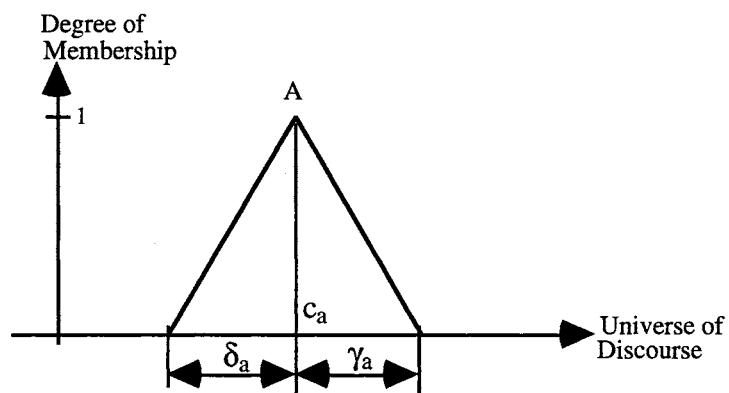


Fig. 2.2 A Typical Fuzzy Number

$$\begin{aligned}
\mu_A(x) &= \frac{x - \delta_a - c_a}{c_a \delta_a} \delta_a \leq x \leq c_a \\
\mu_A(x) &= \frac{x - \gamma_a - c_a}{c_a \gamma_a} c_a \leq x \leq \gamma_a \\
\mu_A(x) &= 0 \quad \text{otherwise}
\end{aligned} \tag{2.1}$$

where A is a particular fuzzy number, c_a is the mean of the fuzzy number, and δ_a and γ_a are the left and right spreads of the fuzzy number, respectively. Accordingly, we designate fuzzy number A by $A = (c_a, \delta_a, \gamma_a)$. If both δ_a and γ_a are zero, the fuzzy number A is a singleton fuzzy number, $A = (c_a, 0, 0)$.

Each fuzzy number of both the bottom-up and top-down fuzzy weight vector is initially given by

$$w_{ij}(0) = (c_{w_{ij}}(0), \delta_{w_{ij}}(0), \gamma_{w_{ij}}(0)) = (1, 1, 1) \tag{2.2}$$

$$z_{ji}(0) = (c_{z_{ji}}(0), \delta_{z_{ji}}(0), \gamma_{z_{ji}}(0)) = (1, 1, 1) \tag{2.3}$$

for $i=1, 2, \dots, m$ and $j=1, 2, \dots, n$. With no F_2^a category nodes being chosen for coding, the nodes are said to be uncommitted. After a category node is selected for coding, it becomes committed.

Fuzzy Subset Functions

Let X represent the universe of discourse, and $\mu_A(x)$ and $\mu_B(x)$ represent membership functions for fuzzy sets (fuzzy numbers) A and B , respectively. Then, based on Zadeh's definition (Zadeh, 1965), fuzzy sets A and B are equal iff $\mu_A(x) = \mu_B(x)$ for every $x \in X$, and A is subset of B iff $\mu_A(x) \leq \mu_B(x)$ for every $x \in X$. This definition of "equal" and "subset" is not truly fuzzy (Kosko, 1986). Suppose that all fuzzy sets discussed here are normal. We define three new relationships of fuzzy set B with respect to fuzzy set A . If the mean and spreads of fuzzy set B are smaller than or equal to those of

fuzzy set A, fuzzy set B is said to be a subset of A. If the mean and spreads of fuzzy set B are larger than those of fuzzy set A, fuzzy set B is a superset of A. If some of mean and spreads of fuzzy set B are smaller than or equal to that of A, and the remainder is larger, fuzzy set B is said to be a mixed set of fuzzy set A. For normal fuzzy sets, this definition of subset subsumes Zadeh's notion of subset. Following the ideas of Kosko (1986), and Carpenter and Grossberg et. al (1991a , 1991b), we present a fuzzy subset function $S(A,B)$ to measure the degree to which fuzzy set B is a subset of A, given by

$$S(A,B) = \frac{|A \otimes B|}{\alpha + |B|} \quad (2.4)$$

where α is an arbitrarily chosen small positive number, $0 < \alpha \ll |B|$; A and B are fuzzy numbers, $A \equiv (c_a, \delta_a, \gamma_a)$ and $B \equiv (c_b, \delta_b, \gamma_b)$; $A \otimes B$ is defined by

$$A \otimes B = (c_a, \delta_a, \gamma_a) \otimes (c_b, \delta_b, \gamma_b) = [\min(c_a, c_b), \min(\delta_a, \delta_b), \min(\gamma_a, \gamma_b)] \quad (2.5)$$

Obviously, $A \otimes B$ is fuzzy subset of both A and B, which we further denote by

$$A \otimes B = (c_{ab}, \delta_{ab}, \gamma_{ab}) \quad (2.6)$$

The fuzzy norm $|\cdot|$ used in (2.4) is defined by

$$|p| = \sum_{i=1}^q |x_i \cdot \mu_p(x_i)| \cdot x_i \quad (2.7)$$

where p is a fuzzy number, q is the number of discrete values x_i in the universe of discourse, and the vertical lines under the summation indicate absolute value. It is easy to prove that this definition satisfies the following norm definition, i.e.,

$$(1) \quad |P| > 0 \quad (2.8)$$

$$(2) \quad |cP| = |c| |P| \quad (2.9)$$

$$(3) \quad |P+Q| \leq |P| + |Q| \quad (2.10)$$

where P and Q are vectors and c is a constant.

With x_i normalized in the interval [0, 1], using (2.7) in (2.4) yields

$$S(A,B) = \frac{\sum_{i=1}^q x_i \cdot \mu_{A \otimes B}(x_i) \cdot x_i}{\alpha + \sum_{i=1}^q x_i \cdot \mu_B(x_i) \cdot x_i} \quad (2.11)$$

As the number of discrete points x_i becomes arbitrarily large, with infinitesimal separations between them, we approach continuous values of x . For this situation, (2.11) becomes:

$$S(A,B) = \frac{\int_{c_{ab}-\delta_{ab}}^{c_{ab}+\gamma_{ab}} x \cdot \mu_{A \otimes B}(x) dx}{\alpha + \int_{c_b-\delta_b}^{c_b+\gamma_b} x \cdot \mu_B(x) dx} \quad (2.12)$$

where \int is the classical integral. When the membership functions of a fuzzy number are triangular as in (2.1), the fuzzy subset function is given by

$$S(A,B) = \frac{\gamma_{ab}^2 + 3c_{ab}(\gamma_{ab} + \delta_{ab}) - \delta_{ab}^2}{6\alpha + (\gamma_b^2 + 3c_b(\gamma_b + \delta_b) - \delta_b^2)} \quad (2.13)$$

If fuzzy numbers are isosceles triangle, or if the spreads of fuzzy numbers are small with respect to their means, $S(A,B)$ in (2.13) reduces

$$S(A,B) = \frac{c_{ab}(\gamma_{ab} + \delta_{ab})}{2\alpha + c_b(\gamma_b + \delta_b)} \quad (2.14)$$

where $0 < S(A,B) < 1$. As $S(A,B)$ increases toward 1.0, a larger portion of B becomes a subset of A. If all of B is a subset of A, then $|A \otimes B| / |B| = 1$. When all of fuzzy sets B and C are subsets of fuzzy set A, $|A \otimes B| = |B|$ and $|A \otimes C| = |C|$, such that $S(A,B) = 1/(\alpha/|B| + 1)$ and $S(A,C) = 1/(\alpha/|C| + 1)$. In this situation, suppose $|B| > |C|$, such that the degree to which

fuzzy set B is equal to fuzzy set A is higher than that of fuzzy set C. The small positive number α is introduced in (2.4) to force $S(A,B) > S(A,C)$.

Fuzzy Signal Transmission

In the proposed FUZART in Fig. 2.1, the transmission of the bottom-up fuzzy signal U_{ij} from the i th F_1^a node to the j th F_2^a node obeys the equation

$$U_{ij} = (c_{u_{ij}}, \delta_{u_{ij}}, \gamma_{u_{ij}}) = (c_{x_i}, \delta_{x_i}, \gamma_{x_i}) \otimes (c_{w_{ij}}, \delta_{w_{ij}}, \gamma_{w_{ij}}) \quad (2.15)$$

where x_i is the i th fuzzy number of the m -dimensional fuzzy output vector x of the field F_1^a , and w_{ij} is the i th fuzzy number of the bottom-up fuzzy weight vector w_j . The transmission of the top-down fuzzy signal D_{ji} from the j th F_2^a node to the i th F_1^a node obeys the equation

$$D_{ji} = (c_{D_{ji}}, \delta_{D_{ji}}, \gamma_{D_{ji}}) = (c_{y_j}, \delta_{y_j}, \gamma_{y_j}) \otimes (c_{z_{ji}}, \delta_{z_{ji}}, \gamma_{z_{ji}}) \quad (2.16)$$

where y_j is the j th fuzzy number of the n -dimensional fuzzy output vector y of the field F_2^a , and z_{ji} is the i th fuzzy number of the top-down fuzzy weight vector z_j . The total fuzzy signal U_j from the field F_1^a to the j th F_2^a node is the fuzzy sum

$$U_j = \sum_{i=1}^m (c_{x_i}, \delta_{x_i}, \gamma_{x_i}) \otimes (c_{w_{ij}}, \delta_{w_{ij}}, \gamma_{w_{ij}}) \quad (2.17)$$

The total fuzzy signal D_i from the field F_2^a to the i th F_1^a node is the fuzzy sum

$$D_i = \sum_{j=1}^n (c_{y_j}, \delta_{y_j}, \gamma_{y_j}) \otimes (c_{z_{ji}}, \delta_{z_{ji}}, \gamma_{z_{ji}}) \quad (2.18)$$

The fuzzy sum is defined by the extension principle (Kosko, 1986)

$$\mu_{A+B}(z) = \max\{\mu_A(x) \wedge \mu_B(y) : z=x+y\} \quad (2.19)$$

where A and B are fuzzy numbers defined by their membership functions $\mu_A(x)$ and $\mu_B(y)$, respectively.

F₁^a Field Activation

The FUZART F₁^a field activation is governed by the activation of the F₀^a field and the F₂^a field. The output of the F₁^a field is given by

$$\mathbf{x} = \begin{cases} (c_p, d_p, \gamma_p) & \text{if } F_2^a \text{ is inactive} \\ (c_p, d_p, \gamma_p) \otimes (c_D, d_D, \gamma_D) & \text{if } F_2^a \text{ is active} \end{cases} \quad (2.20)$$

where $\mathbf{p}=(p_1, p_2, \dots, p_m)$ is a fuzzy input vector or $\mathbf{D}=(D_1, D_2, \dots, D_m)$ is defined by (2.18), and

$$[(c_p, \delta_p, \gamma_p) \otimes (c_D, \delta_D, \gamma_D)]_i \equiv (c_{p_i}, \delta_{p_i}, \gamma_{p_i}) \otimes (c_{D_i}, \delta_{D_i}, \gamma_{D_i}) \quad (2.21)$$

for $i=1,2,\dots,m$. Assume that the F₂^a field operates with winner-take-all dynamics so that only one node ($j=J$) can be active at a time. Let

$$\mathbf{y}_J = (c_{y_J}, \delta_{y_J}, \gamma_{y_J}) = (1, 1, 1) \quad \text{If } j=J \quad (2.22)$$

$$\mathbf{y}_j = (c_{y_j}, \delta_{y_j}, \gamma_{y_j}) = (0, 0, 0) \quad \text{otherwise} \quad (2.23)$$

From (2.18), (2.20), (2.22) and (2.23), we obtain

$$\mathbf{x} = \begin{cases} (c_p, \delta_p, \gamma_p) & \text{if } F_2^a \text{ is inactive} \\ (c_p, \delta_p, \gamma_p) \otimes (c_{z_j}, \delta_{z_j}, \gamma_{z_j}) & \text{if } F_2^a \text{ is active} \end{cases} \quad (2.24)$$

F₂^a Field Category Choice

The total fuzzy signal received by each node of the F₂^a field is the fuzzy sum U_j , as indicated by (2.17). Since the bottom-up and top-down weights are the same in the FUZART, substitute (2.24) into (2.17), we obtain

$$U_j = \sum_{i=1}^m (c_{p_i}, \delta_{p_i}, \gamma_{p_i}) \otimes (c_{w_{ij}}, \delta_{w_{ij}}, \gamma_{w_{ij}}) \quad (2.25)$$

According to (2.17) and (2.19), U_j is a fuzzy number. Following the fuzzy subset function defined by (2.4), the fuzzy category choice function for each input \mathbf{p} and F_2^a node j is defined by

$$T_j(\mathbf{p}) \equiv \frac{\left| \sum_{i=1}^m (c_{p_i}, \delta_{p_i}, \gamma_{p_i}) \otimes (c_{w_{ij}}, \delta_{w_{ij}}, \gamma_{w_{ij}}) \right|}{\alpha + \left| \sum_{i=1}^m (c_{w_{ij}}, \delta_{w_{ij}}, \gamma_{w_{ij}}) \right|} \quad (2.26)$$

where $|\cdot|$ indicates fuzzy norm defined by (2.7), " Σ " indicates fuzzy sum defined by (2.19), and "+" indicates arithmetic addition.

The winner node J in the F_2^a field is selected by

$$T_J(\mathbf{p}) = \max(T_j : j=1, 2, \dots, n) \quad (2.27)$$

J represents a category choice for a given input fuzzy vector \mathbf{p} . Like the fuzzy subset function, $T_J(\mathbf{p})$ reflects the degree to which fuzzy vector \mathbf{w}_j is similar to fuzzy vector \mathbf{p} . If, for some case, more than one T_j is maximal, the lowest index is chosen.

Fuzzy Resonance and Reset

In FUZART, a chosen category fulfills the "fuzzy vigilance criterion" if

$$\frac{\left| \sum_{i=1}^m (c_{p_i}, \delta_{p_i}, \gamma_{p_i}) \otimes (c_{w_{ij}}, \delta_{w_{ij}}, \gamma_{w_{ij}}) \right|}{\left| \sum_{i=1}^m (c_{p_i}, \delta_{p_i}, \gamma_{p_i}) \right|} \geq \rho \quad (2.28)$$

This state is called fuzzy resonance, where ρ is the vigilance parameter, with $0 \leq \rho \leq 1$. Following the occurrence of fuzzy resonance, fuzzy weight update ensues. The top-down fuzzy weight \mathbf{w}_J mismatches the fuzzy input vector if

$$\frac{\left| \sum_{i=1}^m (c_{p_i}, \delta_{p_i}, \gamma_{p_i}) \otimes (c_{w_{ij}}, \delta_{w_{ij}}, \gamma_{w_{ij}}) \right|}{\left| \sum_{i=1}^m (c_{p_i}, \delta_{p_i}, \gamma_{p_i}) \right|} < \rho \quad (2.29)$$

This mismatch causes the FUZART to reset and inhibit the current winner (i.e., set $T_J = (0, 0, 0)$) so that it cannot be chosen for the duration of the input presentation during search. A new node is then selected according to (2.27). The search continues until the fuzzy vigilance criterion (2.28) is satisfied.

Fuzzy Weight Learning

When the J th category node in the F_2^a field is chosen, the fuzzy weights are updated by

$$w_J^{\text{new}} = \beta(\mathbf{p} \otimes w_J^{\text{old}}) + (1-\beta)w_J^{\text{old}} \quad (2.30)$$

provided (2.28) is satisfied, where " $\mathbf{p} \otimes w_J^{\text{old}}$ " is defined by (2.21), "+" is fuzzy sum defined by (2.19), "-" is arithmetic subtraction, and β is a learning parameter with $0 \leq \beta \leq 1$.

Fuzzy Complement Coding

Fuzzy complement coding is a fuzzy input normalization process that represents fuzzy and non-fuzzy features of input vector. Let \mathbf{I} represent the given input pattern with m fuzzy (non-fuzzy) features. The fuzzy complement of \mathbf{I} , denoted by \mathbf{I}^c , represents the non-fuzzy (fuzzy) of each feature,

$$\mathbf{I}^c = (c_{\mathbf{I}^c}, \delta_{\mathbf{I}^c}, \gamma_{\mathbf{I}^c}) = (1-c_{\mathbf{I}}, 1-\delta_{\mathbf{I}}, 1-\gamma_{\mathbf{I}}) \quad (2.31)$$

where "-" indicates arithmetic subtraction. With complement coding, the input vector \mathbf{p} is augmented, such that it becomes $\mathbf{p} = (\mathbf{p}, \mathbf{p}^c)$. With this augmented definition of \mathbf{p} , we note that

$$\sum_{i=1}^{2m} (c_{p_i}, \delta_{p_i}, \gamma_{p_i}) = (m, m, m) \quad (2.32)$$

such that inputs with fuzzy complement coding are automatically normalized.

Theorem: *The FUZART system with fuzzy complement coding forms a finite of number categories. The fuzzy weight learning in (2.30) is stable.*

Proof: A clustering algorithm is said to be stable if it satisfies the following two conditions: (1) No weight vector can "cycle", meaning that no component of a weight vector can assume a value equal to or larger than a value for that component occurring at a previous iteration during the self organization phase; (2) Only a finite number of clusters are formed with infinite presentation of input data during the self organization phase. Without fuzzy complement coding a large number of singleton fuzzy inputs and/or a large number of fuzzy inputs with low mean values would erode monotonically decreasing means and spreads of fuzzy weights, which would result in an arbitrarily large number of categories with arbitrarily large presentation of fuzzy inputs. This can be seen from (2.7), (2.28), and (2.30). Suppose for simplicity that $\beta = 1$ in (2.30), such that fuzzy weight vector \mathbf{w}_j is updated according to

$$\mathbf{w}_j^{\text{new}} = \mathbf{p} \otimes \mathbf{w}_j^{\text{old}}, \quad \beta = 1 \quad (2.33)$$

provided fuzzy vigilance criterion (2.28) is satisfied. Without fuzzy complement coding, with $\beta = 1$, (2.28) states that the fuzzy vigilance criterion is satisfied iff

$$|\mathbf{p} \otimes \mathbf{w}_j^{\text{old}}| \geq |\mathbf{p}| \rho, \quad \beta = 1 \quad (2.34)$$

Assume that after an update by (2.33), the fuzzy vigilance criterion is satisfied, such that

$$|\mathbf{w}_j^{\text{new}}| \geq |\mathbf{p}|\rho, \quad \beta = 1 \quad (2.35)$$

Without fuzzy complement coding, singleton fuzzy inputs and low-mean fuzzy inputs would result in $|\mathbf{p}| \rightarrow 0$, such that $|\mathbf{w}_j^{\text{new}}| \geq 0$. When a large number of singleton fuzzy inputs and/or a large number of low-mean fuzzy inputs are presented to the network,

$$|\mathbf{p} \otimes \mathbf{w}_j^{\text{old}}| = |\mathbf{w}_j^{\text{new}}| \rightarrow 0, \quad \beta = 1 \quad (2.36)$$

As in (2.28), an arbitrarily large number of categories would be formed iff $|\mathbf{w}_j^{\text{new}}| \rightarrow 0$.

With fuzzy complement coding, (2.32), (2.7), and (2.12) can be used to show $|\mathbf{p}| = m^2$.

Then, the fuzzy vigilance criterion is satisfied iff

$$|\mathbf{p} \otimes \mathbf{w}_j^{\text{old}}| \geq m^2 \cdot \rho, \quad \beta = 1 \quad (2.37)$$

If after an update, the fuzzy vigilance criterion is satisfied, then

$$|\mathbf{w}_j^{\text{new}}| \geq m^2 \cdot \rho, \quad \beta = 1 \quad (2.38)$$

In other words, the norm of fuzzy weights with fuzzy complement coding always has a lower bound $m^2 \cdot \rho$, which would prevent formation of an arbitrarily large number of categories with arbitrarily large presentations of fuzzy data. Similar conclusions hold for any value of β , except that more presentations of an input may be needed before stabilization occurs. Moreover, means and spreads of fuzzy weights in (2.30) are updated such that their absolute values always decrease. Therefore, once they have been changed, they cannot return to a previous value. Note that FUZART typically leads to a finite, but slightly different number of recognition categories for different orderings of a given training set.

FUZART Decision Making Criterion

FUZART employs fuzzy patterns for training. During self organization, it clusters the fuzzy patterns, whereas during decision making it labels these clusters with some additional class information. In the FUZART system, the fuzzy category choice function for each input \mathbf{p} and F_2^a node j is given by (2.26). During self organization, the F_2^a field operates with winner-take-all dynamics, so that only the most highly activated category node J is chosen to deliver its output, given by (2.22) and (2.23). After self organization, FUZART encodes all fuzzy patterns as determined by its fuzzy weights. During decision making, however, the output in the F_2^a field may be thought of as the fuzzy class membership value of each pattern. The membership values measure the degree to which a fuzzy input pattern belongs to a category. Accordingly, we define the membership function of the i th fuzzy pattern with respect to category j as

$$v_j(\mathbf{p}_i) = \frac{(T_j(\mathbf{p}_i))^\tau}{\sum_{k=1}^n (T_k(\mathbf{p}_i))^\tau} \quad (2.39)$$

where $0 \leq v_j \leq 1$, \mathbf{p}_i is the i th input fuzzy pattern, and T_k is the category choice function (2.26) for node k in the F_2^a field. The parameter τ is the fuzzy sensitivity parameter, $\tau > 1$, set by the network user, which regulates the amount of fuzziness in the class-membership set. It ranges from absolute hard clustering at $\tau \rightarrow \infty$ to increasingly fuzzier clustering as τ decreases toward 1. When $\tau \rightarrow \infty$, FUZART yields a hard decision, such that a fuzzy input pattern belongs to the most highly activated category with full membership. When $\tau < \infty$, FUZART yields "soft" decisions regarding belonging to a category. The farther a pattern lies from a category, the lower is its membership value for that category. We employ the term "soft decision" to describe a fuzzy pattern corresponding to one or more classes with different degrees of membership. Using soft decisions, FUZART provides a confidence

measure of the classification, which would be very useful in fuzzy environments. We provide examples of this in what follows.

Simulation Results

In this section, we evaluate FUZART using two artificially generated fuzzy data sets. The first fuzzy data set in Simulation 1 consists of sixteen 2-dimensional (x, y) fuzzy vector inputs and was constructed to show how FUZART performs with various values of vigilance parameter ρ . Fig. 2.3 presents a graphical representation of the fuzzy data. Note that it is not intuitively obvious from the fuzzy data where the appropriate clusters should be located. The rectangles in Fig. 2.3 represent the supports of our 2-dimensional fuzzy data. The support of a fuzzy vector \mathbf{p} is defined by

$$\text{support}(\mathbf{p}) = \{\mathbf{x}: \mu_{\mathbf{p}}(\mathbf{x}) > 0\} \quad (2.40)$$

We observe from Fig. 2.3 that it is unclear as to how many clusters should be formed, such that the appropriate number of clusters can be said to be "fuzzy". Using FUZART, the given fuzzy data were clustered four times, with four different vigilance parameters ranging from 0.2 to 0.76 using hard decisions. As shown in Fig. 2.4-2.7 the number of clusters ranges from one with vigilance parameter $\rho=0.2$ to six with vigilance parameter $\rho=0.76$. In each situation, FUZART stabilized into fuzzy data clusters using less than four epochs. The results show that it is important for the user to have some intuition in selecting an appropriate vigilance parameter. Deciding which cluster grouping is most appropriate depends a great deal on the data and the application.

The second fuzzy data set for Simulation 2 consists of nine two-dimensional (x, y) fuzzy vectors, with six of them overlapping. Fig. 2.8 presents a graphical representation of this fuzzy vector set, together with numerical values for the fuzzy data. FUZART was trained using vigilance parameter $\rho=0.6$. It stabilized into clusters using only two epochs. Tables 2.1-2.3 show the simulation results, with different fuzzy sensitivity parameters τ

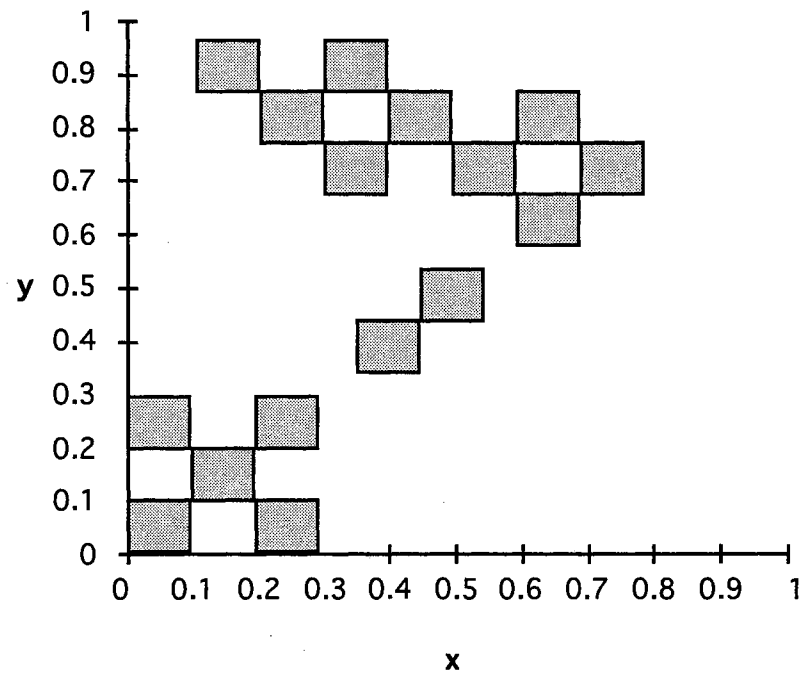


Fig. 2.3 Fuzzy Data Set for Clustering in Simulation 1
 (Shaded Rectangles Represent Fuzzy Support of Input.
 The Spread of Each Fuzzy Number is 0.05)

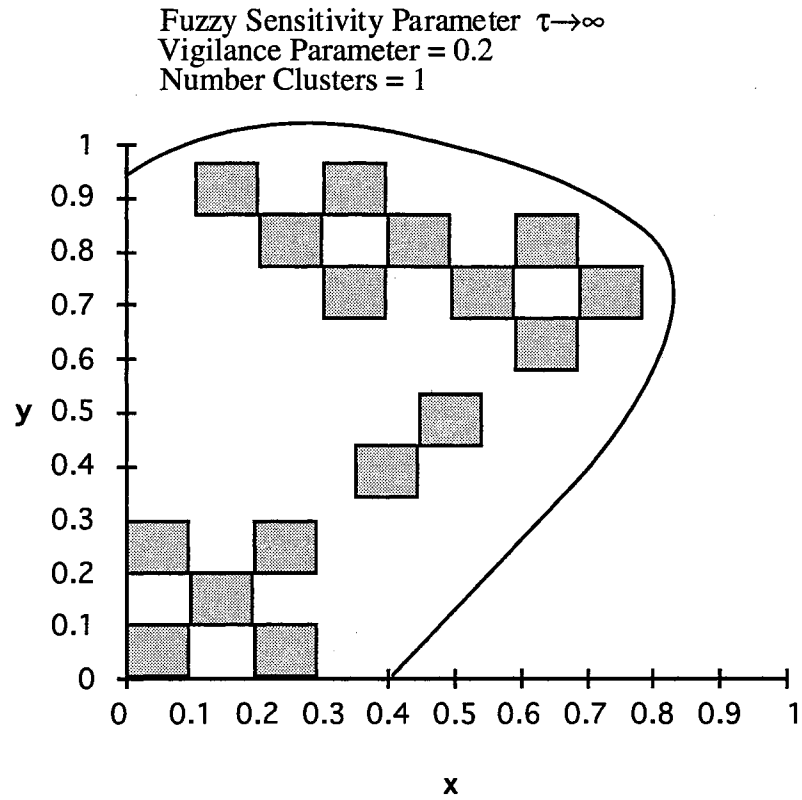


Fig. 2.4 Hard Decision Clustering Performance of FUZART with Vigilance Parameter Equal to 0.2

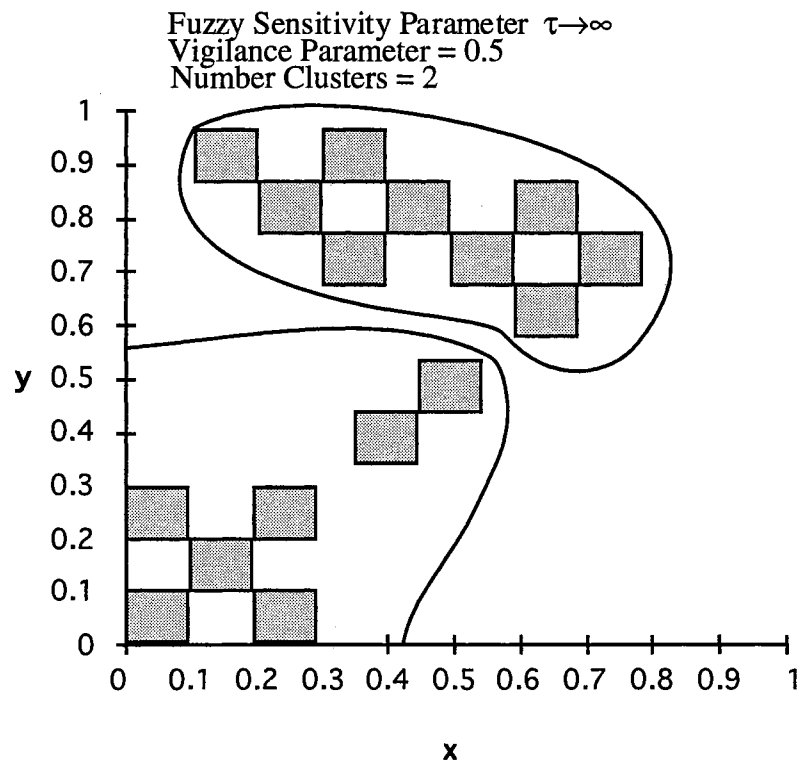


Fig. 2.5 Hard Decision Clustering Performance of FUZART with Vigilance Parameter Equal to 0.5

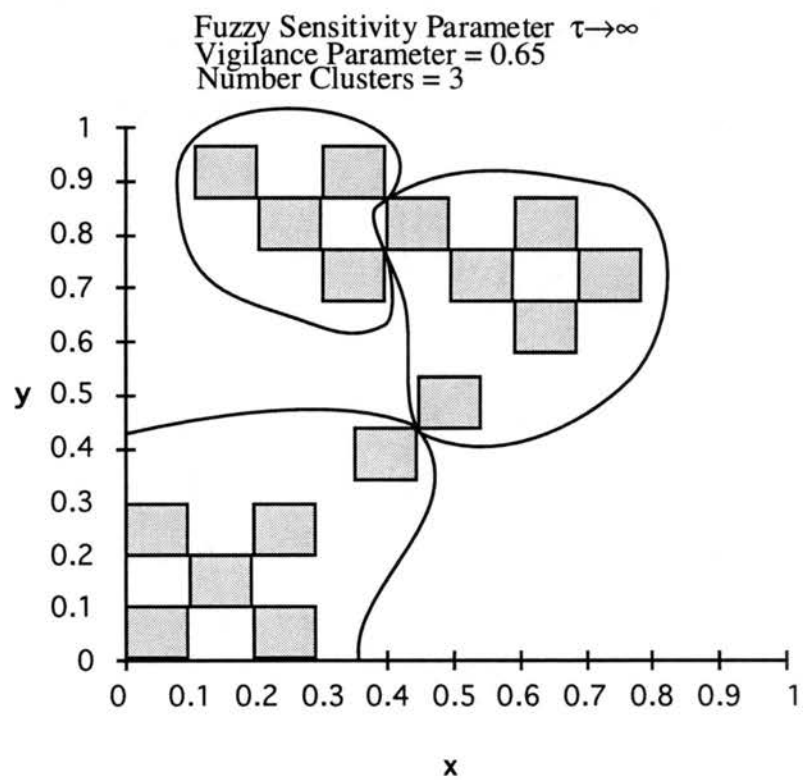


Fig. 2.6 Hard Decision Clustering Performance of FUZART with Vigilance Parameter Equal to 0.65

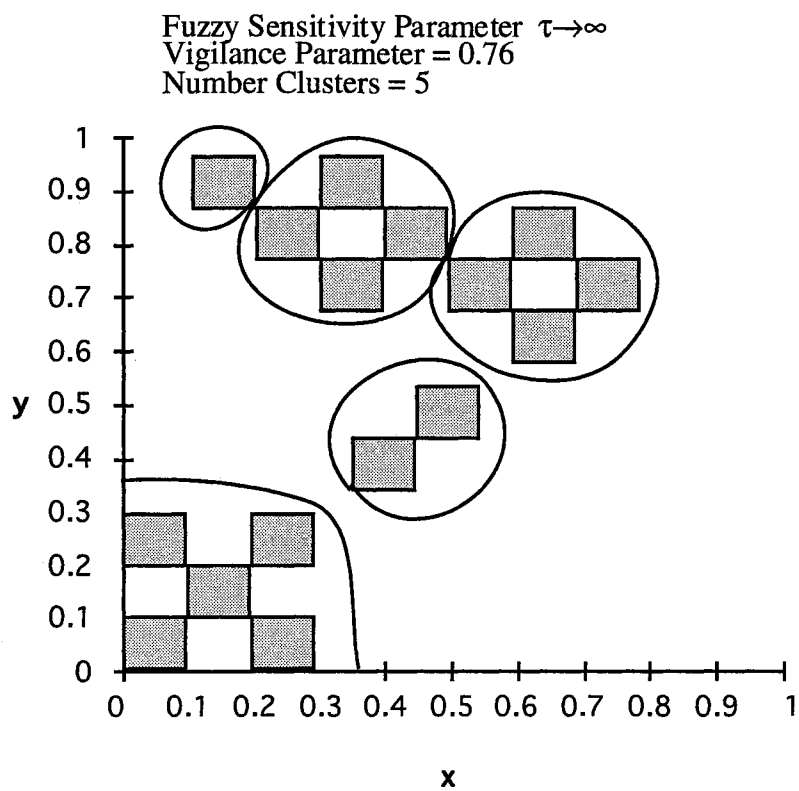
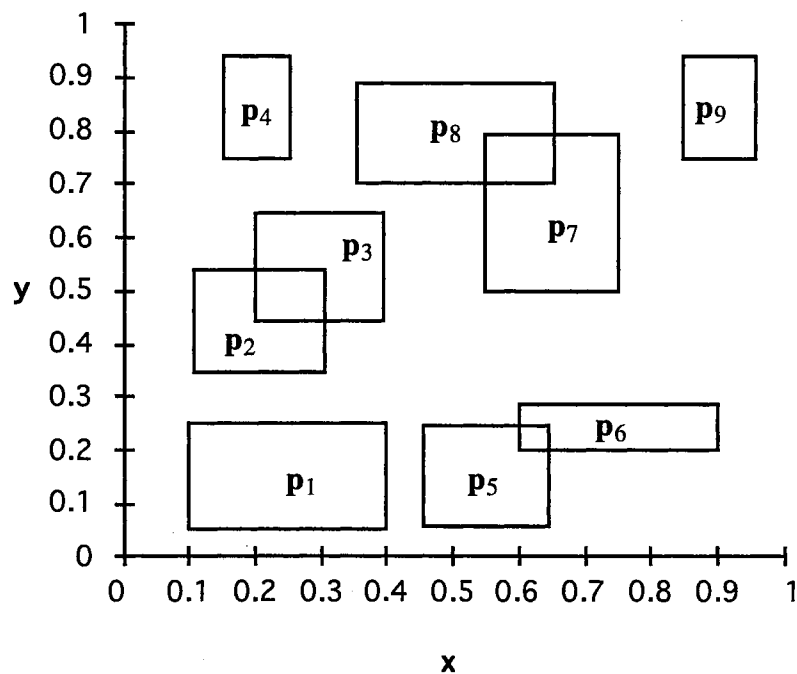


Fig. 2.7 Hard Decision Clustering Performance of FUZART with Vigilance Parameter Equal to 0.76



$p_1=[(0.25,0.15,0.15), (0.15,0.1,0.1)]$ $p_2=[(0.2,0.1,0.1), (0.45,0.1,0.1)]$
 $p_3=[(0.3,0.1,0.1), (0.55,0.1,0.1)]$ $p_4=[(0.2,0.05,0.05), (0.85,0.1,0.1)]$
 $p_5=[(0.55,0.1,0.1), (0.15,0.1,0.1)]$ $p_6=[(0.75,0.15,0.15), (0.25,0.05,0.05)]$
 $p_7=[(0.65,0.1,0.1), (0.65,0.15,0.15)]$ $p_8=[(0.5,0.15,0.15), (0.8,0.1,0.1)]$
 $p_9=[(0.9,0.15,0.15), (0.85,0.05,0.05)]$

Fig. 2.8 Fuzzy Data Set for Fuzzy Clustering in Simulation 2
(Rectangles Represent Fuzzy Support of Input)

Table 2.1 Outputs of FUZART in Simulation 2 with $\rho=0.6$ and $\tau \rightarrow \infty$

Fuzzy Patterns	Fuzzy Outputs (Fuzzy Class Membership Values)			
	v_1 (Cluster 1)	v_2 (Cluster 2)	v_3 (Cluster 3)	v_4 (Cluster 4)
p₁	1.00	0.00	0.00	0.00
p₂	1.00	0.00	0.00	0.00
p₃	1.00	0.00	0.00	0.00
p₄	0.00	1.00	0.00	0.00
p₅	1.00	0.00	0.00	0.00
p₆	0.00	0.00	1.00	0.00
p₇	0.00	0.00	1.00	0.00
p₈	0.00	1.00	0.00	0.00
p₉	0.00	0.00	0.00	1.00

Table 2.2 Outputs of FUZART in Simulation 2 with $\rho=0.6$ and $\tau=4$

Fuzzy Patterns	Fuzzy Outputs (Fuzzy Class Membership Values)			
	v_1 (Cluster 1)	v_2 (Cluster 2)	v_3 (Cluster 3)	v_4 (Cluster 4)
p1	0.75	0.10	0.15	0.00
p2	0.60	0.23	0.15	0.02
p3	0.51	0.27	0.18	0.04
p4	0.19	0.64	0.06	0.11
p5	0.58	0.06	0.33	0.03
p6	0.28	0.04	0.62	0.06
p7	0.21	0.19	0.46	0.14
p8	0.19	0.47	0.19	0.15
p9	0.03	0.20	0.18	0.59

Table 2.3 Outputs of FUZART in Simulation 2 with $\rho=0.6$ and $\tau=1$

Fuzzy Patterns	Fuzzy Outputs (Fuzzy Class Membership Values)			
	v_1 (Cluster 1)	v_2 (Cluster 2)	v_3 (Cluster 3)	v_4 (Cluster 4)
p₁	0.39	0.23	0.26	0.12
p₂	0.34	0.27	0.24	0.15
p₃	0.32	0.27	0.24	0.17
p₄	0.25	0.34	0.19	0.22
p₅	0.34	0.20	0.30	0.16
p₆	0.28	0.17	0.34	0.21
p₇	0.24	0.24	0.30	0.22
p₈	0.24	0.30	0.24	0.22
p₉	0.16	0.25	0.25	0.34

ranging from $\tau \rightarrow \infty$ to $\tau=1$. Each entry under the column v_i corresponds to cluster i 's membership value produced by FUZART. When $\tau \rightarrow \infty$, shown in Table 2.1, hard decisions for membership clustering result, such that fuzzy patterns \mathbf{p}_1 , \mathbf{p}_2 , \mathbf{p}_3 , and \mathbf{p}_5 belong to Cluster 1, fuzzy patterns \mathbf{p}_4 and \mathbf{p}_8 belong to Cluster 2, fuzzy patterns \mathbf{p}_6 and \mathbf{p}_7 belong to Cluster 3, and fuzzy pattern \mathbf{p}_9 belongs to Cluster 4. FUZART also provides soft clustering decisions. Table 2.2 shows results from a soft decision simulation with $\tau=4$. As an example, consider the entries in the 3rd row for \mathbf{p}_3 . It can be observed that the degrees to which pattern \mathbf{p}_3 fits within Clusters 1-4 are 0.51, 0.27, 0.18, and 0.04, respectively. Table 2.3 shows soft decision simulation results with $\tau=1$. Consider again the entries in the 3rd row for \mathbf{p}_3 . It can be observed that the degrees to which pattern \mathbf{p}_3 fits within Clusters 1 - 4 become 0.32, 0.27, 0.24, and 0.17, respectively. By comparison with results in Tables 2.1-2.3, we can see that τ regulates the amount of fuzziness in the cluster-membership set. In particular, we note that for $\tau=1$, while the degree of membership of \mathbf{p}_3 in Cluster 1 is the largest of all cluster memberships, membership degrees in Clusters 2 and 3 are close to that of Cluster 1. Thus while \mathbf{p}_3 "best fits" Cluster 1, it fits Clusters 2 and 3 almost well. It appears that this soft decision mechanism provides higher level decision and information processing with additional valuable information.

Summary

An unsupervised fuzzy neural network for fuzzy patterns, termed FUZART, based on Adaptive Resonance Theory networks has been proposed. FUZART employs the two stages of self-organization and decision making. It accepts fuzzy input patterns and provides output decisions in terms of membership values. FUZART has the ability to learn on-line using only a few training epochs. Simulation results demonstrate the ability of FUZART to provide reasonable clustering decisions for fuzzy patterns, but there exists room for further investigation of FUZART. For example, the FUZART network and soft

decision mechanisms proposed here might be extended for supervised neural networks, such that the performance of supervised neural network would be improved. In the next chapter, these issues are addressed.

CHAPTER III

A SUPERVISED FUZZY NEURAL NETWORK ARCHITECTURE FOR FUZZY CONTROL AND CLASSIFICATION

Introduction

Research in supervised neural networks has become very active recently, and a number of interesting approaches have been proposed (Hayashi and Buckley et al., 1993; Ishibuchi et al., 1993, 1994a, 1994b; Keller and Tahani, 1992a, 1992b). Keller and Tahani (1992a, 1992b) propose a supervised fuzzy neural network which discretizes a single fuzzy set and uses the membership values of discrete elements as inputs of the neural networks. Each node in the input layer of their fuzzy neural network represents a membership value of the fuzzy set for a particular element. Membership values are propagated through the neural network using real weights. Ishibuchi, et al. (1993, 1994a, 1994b) proposed a supervised fuzzy neural network using level sets of a fuzzy set. Each node in the input layer of their fuzzy neural network represents a fuzzy number, and fuzzy inputs are propagated through the neural network with learning of the real weights. Hayashi et al. (1993) propose a supervised fuzzy neural network that can propagate fuzzy inputs using fuzzy weights, but no numerical results were presented in their work. All of these approaches use standard or modified back propagation (BP) algorithms. However, a fundamental difficulty with the BP algorithm used for fuzzy neural networks is that it does not allow new fuzzy data to be added into the fuzzy neural network without retraining. In contrast to employing BP algorithm-based approaches for dealing with fuzzy data, we propose herein a new supervised fuzzy neural network called FUZAMP, based on Adaptive Resonance Theory networks ART and ARTMAP proposed by Carpenter and

Grossberg et al. (1991a, 1991b), which can realize on-line supervised learning of recognition categories. FUZAMP is a natural extension of FUZART, described in Chapter II and propagates fuzzy data using fuzzy weights through three layers of network nodes, with each node representing a fuzzy number. The substantial advantages of ARTMAP compared to BP networks have been shown by Feng and Hoberock (1992) in a manufacturing queuing problem with binary input data. Accordingly, we seek a new fuzzy neural network based on ARTMAP that can achieve results of fuzzy inference and fuzzy classification comparable to BP algorithms while using only a few training epochs. This is in contrast to BP neural networks, which may need large amounts of training epochs even for problems of small size, and which also may have local minimum problems (Hayashi and Buckley et al., 1993; Ishibuchi et al., 1993, 1994a, 1994b; Keller and Tahani, 1992a, 1992b). It is desired to develop this new fuzzy neural network to especially handle problems of large size with fast convergence. In what follows, we first describe desired properties of a supervised fuzzy neural network. Following this, we discuss the algorithm of our new supervised fuzzy neural network, FUZAMP, which is well-suited for control and classification problems. Finally, simulation results are given to demonstrate the performance of FUZAMP.

Desired Properties of A Supervised Fuzzy Neural Network

In employing fuzzy neural networks to implement fuzzy rules for fuzzy control systems, there are several properties that a fuzzy neural network should possess. These properties include:

- 1. On-Line Learning Capability:* A fuzzy neural network should be able to learn new rules and refine existing rules quickly without destroying old information. Many fuzzy neural networks use off-line learning (Hayashi and Buckley et al., 1993; Ishibuchi et al., 1993, 1994a, 1994b; Keller and Tahani, 1992a, 1992b). Each time new information is added to

such fuzzy neural networks, retraining of the system is required using both old and new information, which might result in longer training times and require large memory.

- 2. *Short Training Time:* One of the most important features of fuzzy neural network learning algorithms is the amount of training time required. Short training times are much preferred. Many fuzzy neural networks (Hayashi and Buckley et al., 1993; Ishibuchi et al., 1993, 1994a, 1994b; Keller and Tahani, 1992a, 1992b) require from hundreds to thousands of training epochs, which appear excessive.
3. *Good Fitting Capability:* When fuzzy targets are removed from the trained fuzzy neural network, the outputs produced by the network should be the same as the training targets corresponding to the same fuzzy input patterns in the training set. Some fuzzy neural networks (Ishibuchi et al., 1993, 1994a, 1994b) do not have such good fitting capability.
4. *Good Robustness:* In some applications it is desired that fuzzy outputs produced by a fuzzy neural network be insensitive to changes in fuzzy inputs, for which the fuzzy neural network is said to be robust to changes in fuzzy inputs. Some fuzzy neural networks (Ishibuchi et al., 1993, 1994a, 1994b) are not robust in this sense.
5. *Good Generalization:* A fuzzy neural network should have ability to interpolate the missing fuzzy rules using a limit number of available fuzzy rules. Many fuzzy neural networks do not have such a property (Hayashi and Buckley et al., 1993; Keller and Tahani, 1992a, 1992b).

In the simulation examples at the end of this chapter, we demonstrate that FUZAMP possesses all five of these properties to a high degree.

FUZAMP Algorithm

Fig. 3.1 illustrates the architecture of the new FUZAMP. It consists of a FUZART network described in Chapter II, together with a fuzzified mapping network. The fuzzified mapping network between FUZART and target field F_1^b establishes an interaction of fuzzy input and fuzzy target patterns. Fuzzy inputs presented in the F_0^a field are associated with

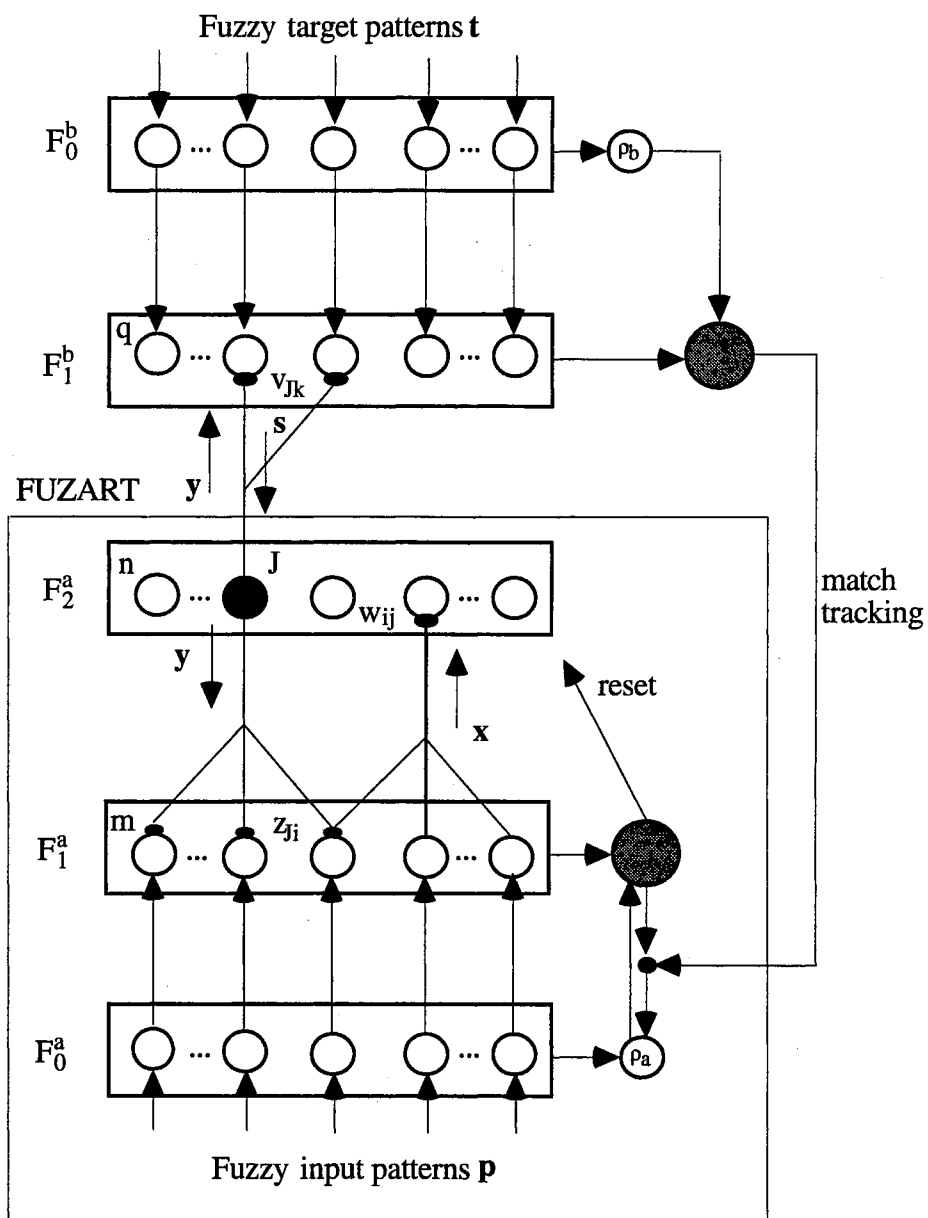


Fig. 3.1 The Architecture of FUZAMP

[Adapted from ARTMAP (Carpenter, Grossberg, et al., 1991a)]

fuzzy targets presented in the F_0^b field. As for FUZART, FUZAMP can be applied to both non-singleton fuzzy input data and singleton fuzzy input data (analog and binary data), and their combinations. The target patterns in FUZAMP can also be singleton, non-singleton, or combined fuzzy data. For singleton fuzzy data, FUZAMP is consistent with fuzzy ARTMAP introduced by Carpenter and Grossberg et al. (1991a, 1991b). In this sense, FUZAMP proposed in this chapter can be considered as an extension of fuzzy ARTMAP to the more general case of non-singleton fuzzy data.

As illustrated in Fig. 3.1, q designates number of nodes in the F_1^b field, and each node in the F_1^b field represents a fuzzy number and is designated by the index k . When a fuzzy prediction v_{jk} by FUZART is not confirmed at F_1^b by fuzzy target patterns, inhibition by field F_1^b activation induces the fuzzy match tracking process. Fuzzy match tracking raises FUZART vigilance parameter ρ_a by the minimal amount required for a fuzzy prediction by FUZART to be confirmed at F_1^b by fuzzy target patterns. This triggers a FUZART search that leads to activation of either a FUZART category that correctly predicts fuzzy output pattern or to a previously uncommitted FUZART category node.

F_1^b Field Activation

Let $\mathbf{t} = (t_1, t_2, \dots, t_q)$ denote the q -dimensional fuzzy target vector over the field F_0^b , $\mathbf{s} = (s_1, s_2, \dots, s_q)$ denote the q -dimensional fuzzy output vector of the field F_1^b , and $\mathbf{v}_j = (v_{j1}, v_{j2}, \dots, v_{jq})$ denote the mapping fuzzy weights of each node over the field F_1^b , as illustrated in Fig. 3.1. The mapping field F_1^b activation is governed by the activation of the field F_0^b and field F_2^a . The output of the field F_1^b is given by

$$\mathbf{s} = (\mathbf{c}_t, \delta_t, \gamma_t) \otimes (\mathbf{c}_{\mathbf{v}_j}, \delta_{\mathbf{v}_j}, \gamma_{\mathbf{v}_j}) \quad \text{if the } j\text{th } F_2^a \text{ node is active and } F_0^b \text{ is active} \quad (3.1)$$

$$\mathbf{s} = (\mathbf{c}_t, \delta_t, \gamma_t) \quad \text{if } F_2^a \text{ is inactive and } F_0^b \text{ is active} \quad (3.2)$$

$$\mathbf{s} = (\mathbf{c}_{\mathbf{v}_j}, \delta_{\mathbf{v}_j}, \gamma_{\mathbf{v}_j}) \quad \text{if the } j\text{th } F_2^a \text{ node is active and } F_0^b \text{ is inactive} \quad (3.3)$$

$$s = (\mathbf{0}, \mathbf{0}, \mathbf{0}) \quad \text{if both } F_2^a \text{ and } F_0^b \text{ are inactive} \quad (3.4)$$

Fuzzy Match Tracking

When the fuzzy input and fuzzy target are first presented, FUZART vigilance parameter ρ_a is equal to a baseline vigilance $\bar{\rho}_a$. If the current winner node can not meet the vigilance criterion in F_1^b , i.e. if

$$\frac{\left| \sum_{i=1}^q (c_{t_i}, \delta_{t_i}, \gamma_{t_i}) \otimes (c_{v_{j_i}}, \delta_{v_{j_i}}, \gamma_{v_{j_i}}) \right|}{\left| \sum_{i=1}^q (c_{t_i}, \delta_{t_i}, \gamma_{t_i}) \right|} < \rho_b \quad (3.5)$$

where ρ_b is the target vigilance parameter, shown in Fig. 3.1, then fuzzy match tracking raises FUZART vigilance ρ_a by a certain small amount and searches for a new winner node in the F_2^a field. Fuzzy match tracking continues until fuzzy resonance occurs, i.e.,

$$\frac{\left| \sum_{i=1}^q (c_{t_i}, \delta_{t_i}, \gamma_{t_i}) \otimes (c_{v_{j_i}}, \delta_{v_{j_i}}, \gamma_{v_{j_i}}) \right|}{\left| \sum_{i=1}^q (c_{t_i}, \delta_{t_i}, \gamma_{t_i}) \right|} \geq \rho_b \quad (3.6)$$

Target vigilance parameter ρ_b is set to one for many-to-one mapping.

Once the J th category node in the F_2^a field is chosen to associate the fuzzy input pattern of the F_0^a field with the fuzzy target pattern of the F_0^b field, the mapping fuzzy weights are updated by

$$\mathbf{v}_J = \mathbf{t} \quad (3.7)$$

Note that the FUZAMP system is stable since FUZAMP consists of a stable FUZART system and the fuzzified mapping network with a finite number of targets.

Fuzzy Decision

FUZAMP can be thought of as operating in two modes, namely, training and testing. During training, a collection of fuzzy input patterns and fuzzy target patterns are repeatedly presented to the network using "winner-take-all" strategy until fuzzy weights converge to constants. During testing, however, FUZAMP employs a decision making criterion as in (2.39) to integrate distributed activities across coded categories. In the FUZART system, the membership function $v_j(\mathbf{p}_i)$ of the i th fuzzy pattern to class j is defined by (2.39). The output of the F_1^b field in the FUZAMP system during testing is defined by

$$d_k = \sum_{j=1}^n v_j(\mathbf{p}_i) \cdot v_{jk} \quad (3.8)$$

where $d_k = (c_{d_k}, \delta_{d_k}, \gamma_{d_k})$, $v_{jk} = (c_{v_{jk}}, \delta_{v_{jk}}, \gamma_{d_{jk}})$, and \sum is fuzzy sum. The multiplication of fuzzy numbers under the summation sign is defined by the extension principle (Kosko, 1986)

$$\mu_{kA}(z) = \max\{\mu_A(x): z=kx\} \quad (3.9)$$

where A is a fuzzy number defined by its membership function $\mu_A(x)$ and k is a constant. At $\tau = \infty$ in (2.39), a fuzzy input pattern belongs to the most highly activated category with full membership. In such a case FUZAMP employs the hard decision criterion. As τ becomes smaller, FUZAMP uses more information from the relative F_2^a activities. In such cases, FUZAMP employs soft decision criteria.

Simulation and Results

In order to illustrate the performance of the proposed FUZAMP, simulation results are presented in this section. We present four simulations illuminating the capability of FUZAMP as a fuzzy inference engine and three simulations illuminating the fuzzy

classification capability of FUZAMP. We demonstrate in these simulations that FUZAMP possesses all the properties listed earlier of a good fuzzy neural network.

Simulations 1-4: Implementation of Fuzzy Rules for Fuzzy Control Systems

In Simulation 1, assume that the following two fuzzy rules are given for constructing a fuzzy control system:

IF x is small THEN y is small

IF x is large THEN y is large

where "small" and "large" are defined by the membership functions shown in Fig. 3.2. We wish to train FUZAMP to learn these two rules. From the above two fuzzy rules, we can obtain the following fuzzy training data:

$$\{(x,y)\} = \{(small, small), (large, large)\} \quad (3.10)$$

where x is the fuzzy input pattern (data), and y is the fuzzy target pattern (data). After only one epoch of training with fuzzy complement coding, baseline vigilance parameter $\rho_a=0.5$, target vigilance parameter $\rho_b=1$, fuzzy sensitivity parameter $\tau \rightarrow \infty$, and the number of FUZART category nodes equal to 2, FUZAMP was found to be stabilized. The trained FUZAMP was tested with a variety of fuzzy input data using the hard decision criterion. We identified those fuzzy input data as "small", "large", "very small", "very large", "shifted small", "shifted large", "singleton small", and "singleton large". Fuzzy input "small" and "large" are the same as those illustrated in Fig. 3.2. Fuzzy input "very small", "very large", "shifted small", "shifted large", "singleton small", and "singleton large" are shown in Fig. 3.3, Fig. 3.4, and Fig. 3.5, respectively. Fuzzy outputs from the trained FUZAMP are shown in Fig. 3.6, with "small" corresponding to fuzzy inputs "small", "shifted small", "very small", "singleton small" and with "large" corresponding to fuzzy inputs "large", "shifted large", "very large", "singleton large". FUZAMP maps the trained

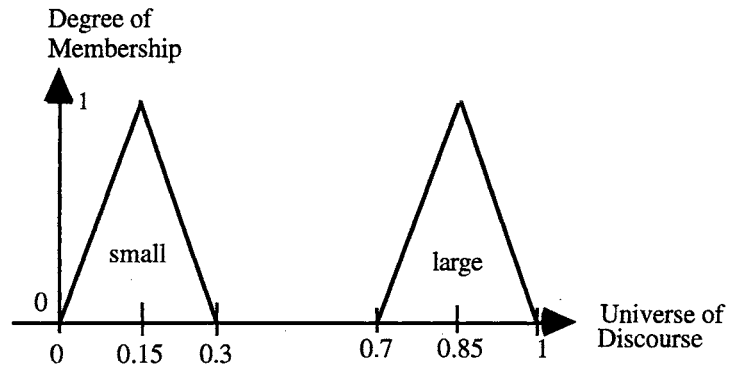


Fig. 3.2 Membership Functions for "small" and "large"

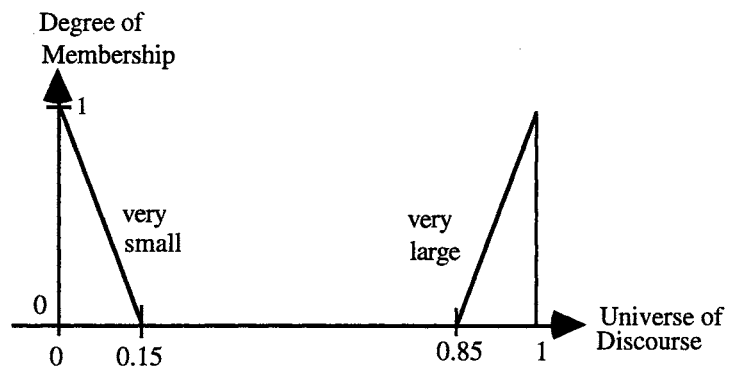


Fig. 3.3 Membership Functions for
"very small" and "very large"

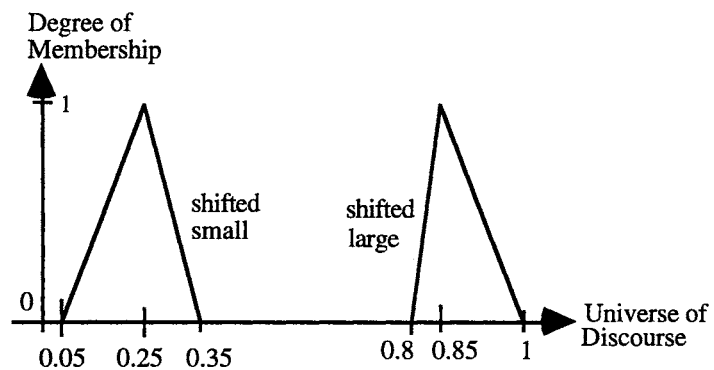


Fig. 3.4 Membership Functions
for "shifted small" and "shifted large"

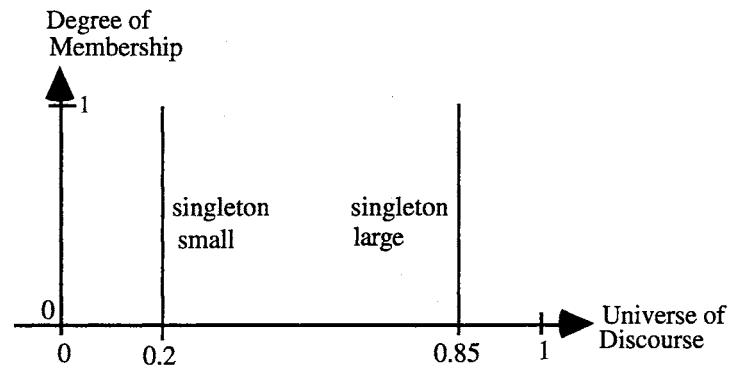


Fig. 3.5 Membership Functions
for "singleton small" and "singleton large"

单个的

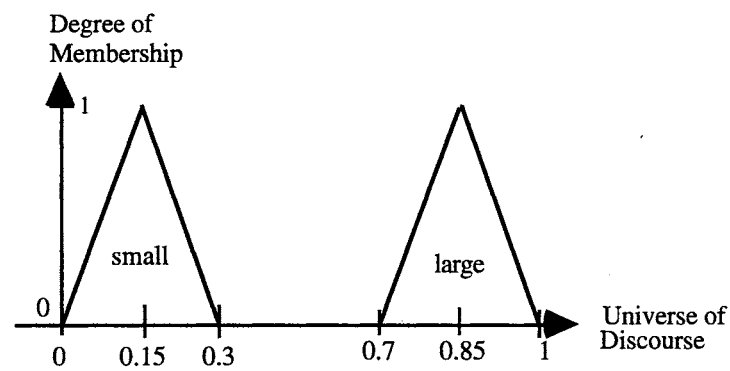


Fig. 3.6 Fuzzy Outputs From
the Trained FUZAMP in Simulation 1

input $\{x\} = \{\text{small, large}\}$ to corresponding trained output $\{y\} = \{\text{small, large}\}$ through the relationship learned during training. Moreover, the trained FUZAMP also maps the shifted input and singleton input to the trained output. The result matches our expectation. FUZAMP demonstrates robustness to changes in fuzzy inputs and has 100% fitting capability with respect to the trained inputs.

In Simulation 2, we consider the following four fuzzy rules

$$\begin{aligned}
 &\text{IF } x \text{ is small and } y \text{ is small THEN } z \text{ is small} \\
 &\text{IF } x \text{ is large and } y \text{ is large THEN } z \text{ is small} \\
 &\text{IF } x \text{ is small and } y \text{ is large THEN } z \text{ is large} \\
 &\text{IF } x \text{ is large and } y \text{ is small THEN } z \text{ is large}
 \end{aligned} \tag{3.11}$$

The membership functions for "small" and "large" are given in Fig. 3.7. The fuzzy rules in (3.11) constitute a typical linearly non-separable problem. From the fuzzy rules in (3.11), FUZAMP was trained via fuzzy input pattern $\{(x,y)\} = \{(\text{small,small}), (\text{large,large}), (\text{small,large}), (\text{large,small})\}$ and fuzzy target pattern $\{z\} = \{(\text{small}), (\text{small}), (\text{large}), (\text{large})\}$. After only one epoch of training, with fuzzy complement coding, baseline vigilance parameter $\rho_a=0.6$, target vigilance parameter $\rho_b=1$, fuzzy sensitivity parameter $\tau \rightarrow \infty$, and the number of FUZART category nodes equal to 4, FUZAMP was found to be stabilized. FUZAMP was tested with the training fuzzy input patterns and a variety of fuzzy input test patterns to check its performance using the hard decision criterion. Fig. 3.8-3.10 illustrate some of the fuzzy input test patterns. Fuzzy outputs from the F_0^b field from the trained FUZAMP are shown in Fig. 3.11, with the output labeled "small" corresponding to the first two input combinations in (3.11) and the output labeled "large" corresponding to the last two input combinations in (3.11). The inputs were taken as all possible combinations from Fig. 3.7-3.10. Comparing Fig. 3.11 with Fig. 3.7, one can see that the fuzzy outputs over the F_0^b field correspond to fuzzy test inputs and match our expectation. The trained FUZAMP generalized the fuzzy relationships implied in the fuzzy

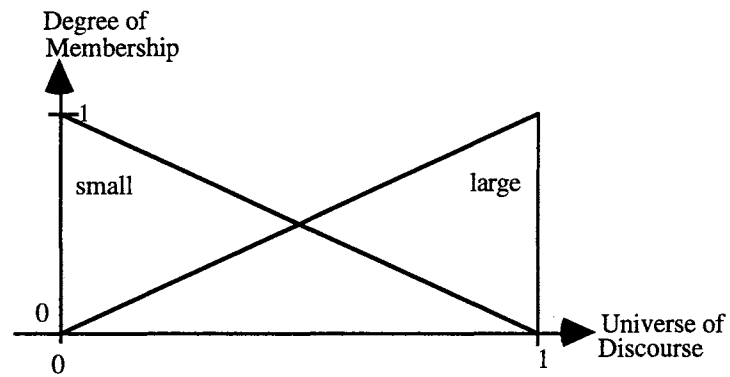


Fig. 3.7 Membership Functions
for "small" and "large" in Simulation 2

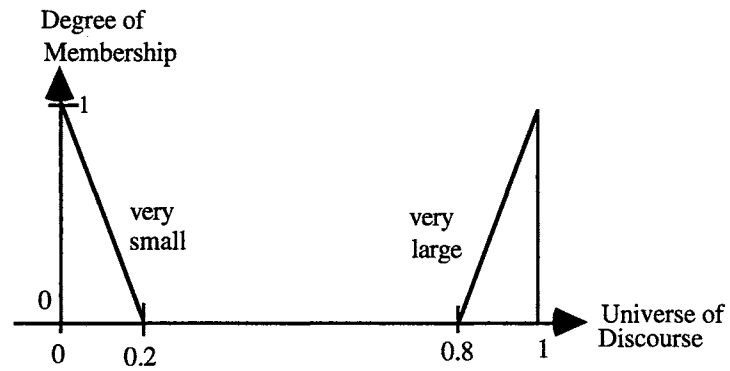


Fig. 3.8 Fuzzy Input Test Data
for "very small" and "very large"

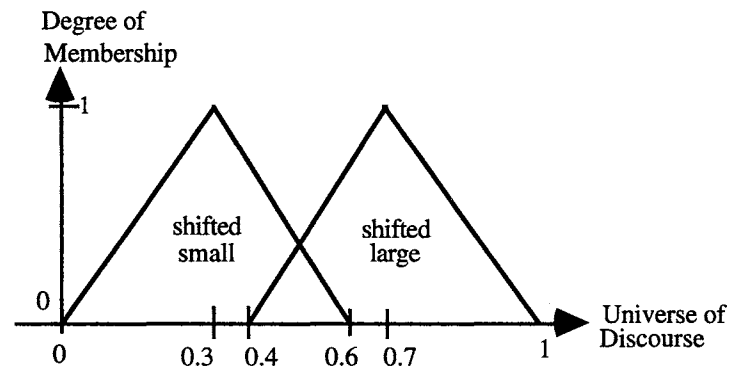


Fig. 3.9 Fuzzy Input Test Data for
"shifted small" and "shifted large"

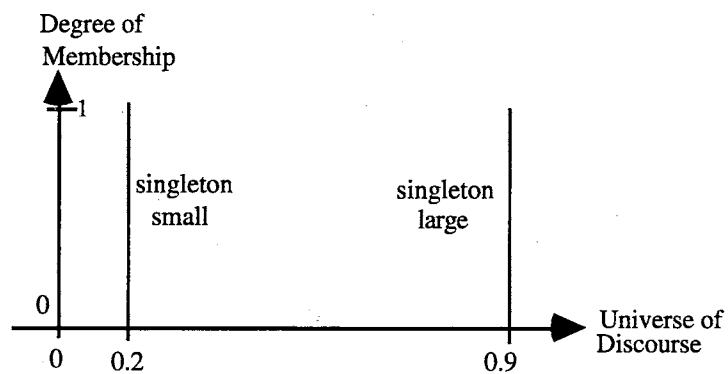


Fig. 3.10 Fuzzy Input Test Data for "singleton small" and "singleton large"

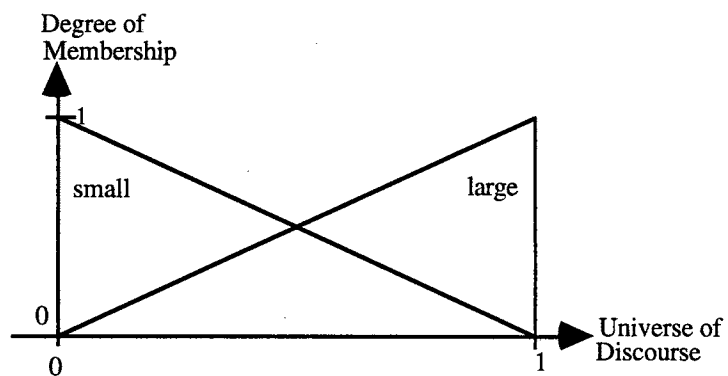


Fig. 3.11 Fuzzy Outputs for the Trained FUZAMP in Simulation 2

rules very well. In particular, FUZAMP maps fuzzy inputs with shape changes to the desired fuzzy output. Again, we find that FUZAMP has 100% fitting capability to the trained data and is robust to changes in input patterns.

In Simulation 3, we trained FUZAMP with three fuzzy input-target pairs (fuzzy rules) shown in Fig. 3.12. In this figure, x represents the fuzzy input; y represents the fuzzy target; and each rectangle illustrates fuzzy support (2.43) of a fuzzy input and its corresponding fuzzy target, each having symmetric triangular membership functions. After only one epoch of training, with fuzzy complement coding, baseline vigilance parameter $\rho_a=0.8$, target vigilance parameter $\rho_b=1$, and the number of FUZART category nodes equal to 3, FUZAMP was found to be stabilized. FUZAMP was then tested using only the input patterns of the three training patterns, plus two novel fuzzy input patterns, to check its performance using the soft decision criterion (3.8). Fig. 3.13 shows the supports of these fuzzy inputs and the corresponding fuzzy outputs produced by FUZAMP, with solid rectangles illustrating the inputs from the training patterns and their corresponding outputs and dotted rectangles illustrating the novel fuzzy inputs and their corresponding fuzzy outputs. Fuzzy sensitivity parameter τ in (2.39) was selected as 8, such that for each fuzzy input pattern on the training set, FUZAMP produced a desired fuzzy output with a membership value of at least 0.99. Note that the solid rectangles perfectly match those of the training set in Fig. 3.12, and that the dotted rectangles nicely fit in between the solid rectangles, following the trend set by the training patterns. These results demonstrate good generalization capability of FUZAMP for the two novel fuzzy inputs and excellent fitting capability for the three training inputs.

In Simulation 4, we assume an incomplete rule table, shown in Table 3.1. The linguistic values in the table are illustrated in Fig. 3.14. As an example for reading this table, consider the entry in the 3rd row, 1st column. The rule is read: "If X is Very Small and Y is Medium, then Z is Large". We trained FUZAMP with the nine input-target pairs obtained from the given fuzzy rules in Table 3.1. After only two epochs of training, with

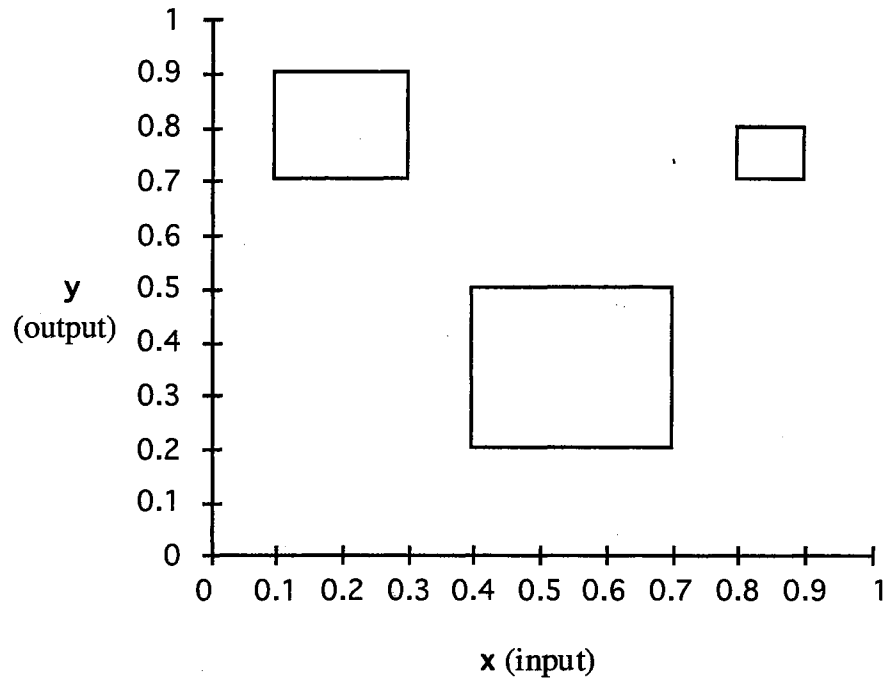


Fig. 3.12 Fuzzy Trained Data in Simulation 3

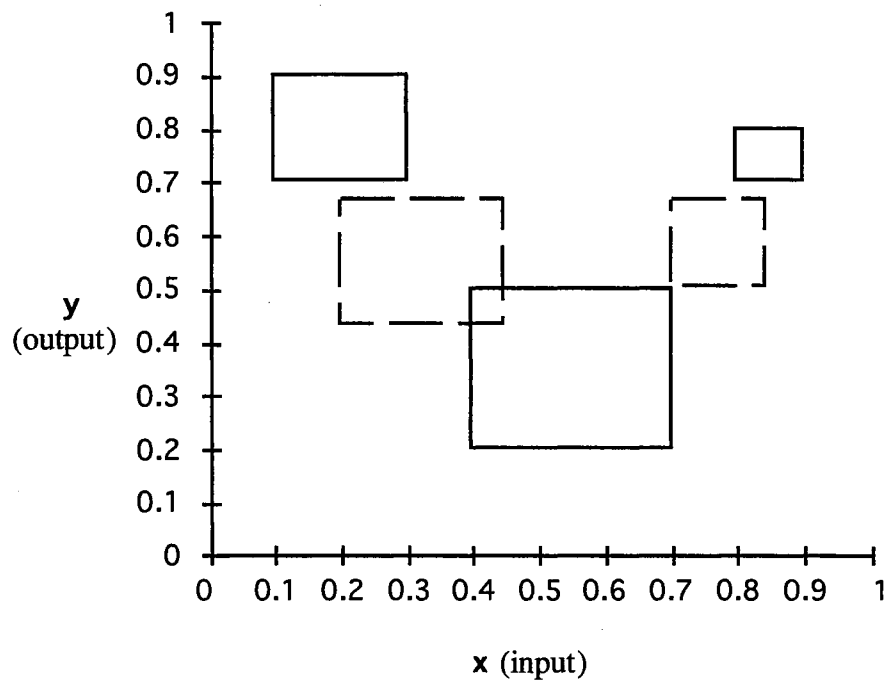


Fig. 3.13 Generalization of the Trained FUZAMP in Simulation 3

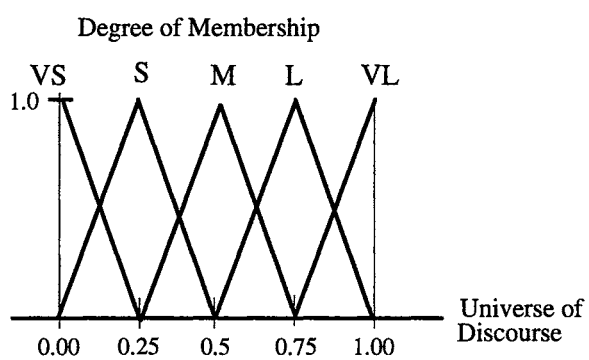


Fig. 3.14 Fuzzy Sets for Five Linguistic Values
(VS: Very Small; S: Small; M: Medium; L: Large; VL: Very Large)

Table 3.1 Available Fuzzy Rules In Simulation 4
(Output Z Corresponding to Inputs X and Y)

X → Y ↓	VS	S	M	L	VL
VS	VL		L		M
S					
M	L		S		VS
L					
VL	M		VS		VS

fuzzy complement coding, baseline vigilance parameter $\rho_a=0.9$, target vigilance parameter $\rho_b=1$, and the number of FUZART category nodes equal to 7, FUZAMP was found to be stabilized. FUZAMP was then employed to interpolate missing fuzzy rules in Table 3.1 using the soft decision criterion. Fuzzy sensitivity parameter τ was selected as 8, such that for each fuzzy input on the training set, FUZAMP produced a desired fuzzy output with a membership value of at least 0.99. We then presented the trained FUZAMP with fuzzy input pairs (x, y) corresponding to the antecedents of the missing fuzzy rules. Each output (z) from FUZAMP was compared with each of the five linguistic values using category choice function and fuzzy vigilance criterion as in (2.26) and (2.28). In (2.26) and (2.28), \mathbf{p} was replaced by the output described herein; \mathbf{w} was replaced by each linguistic value; and \mathbf{m} was equal to one. The linguistic value with a maximum value of category choice function was chosen as the consequent of the missing fuzzy rule. If the category choice function had the same value for different linguistic values, the linguistic value with a maximum value of fuzzy resonance criterion was chosen as the consequent of the missing fuzzy rule. In this way, the consequents of all missing fuzzy rules were determined as shown in italics in Table 3.2. As an example, consider the result in the 4th row, 4th column of Table 3.2. The interpolated rule is: "If X is Large and Y is Large, then Z is Small". We suggest the reader cover Table 3.2 and use human intuition to complete the missing rule results in Table 3.1. Then by comparing the reader's results with those in Table 3.2, the reader may see that the italicized results match human intuition in reasonable fashion. We conclude that FUZAMP has very reasonable generalization capability, with good capability of interpolation from available fuzzy rules.

It should be noted that although BP based fuzzy neural networks have the capability of fuzzy inference for fuzzy data similar to that demonstrated here by FUZAMP, the training time for the FUZAMP is much less than for BP fuzzy neural networks. For example, in Hayashi and Buckley et al. (1993); Ishibuchi et al. (1993, 1994a, 1994b); Keller and Tahani (1992a, 1992b), typically, the number of training epochs required for

Table 3.2 Fuzzy Rule Table Completed by FUZAMP In Simulation 4
(Output Z Corresponding to Inputs X and Y)

X → Y ↓	VS	S	M	L	VL
VS	<i>VL</i>	<i>VL</i>	<i>L</i>	<i>L</i>	<i>M</i>
S	<i>VL</i>	<i>L</i>	<i>M</i>	<i>M</i>	<i>S</i>
M	<i>L</i>	<i>M</i>	<i>S</i>	<i>S</i>	<i>VS</i>
L	<i>L</i>	<i>M</i>	<i>S</i>	<i>S</i>	<i>VS</i>
VL	<i>M</i>	<i>S</i>	<i>VS</i>	<i>VS</i>	<i>VS</i>

stabilizing BP fuzzy neural networks on fuzzy data similar to that in the simulations above ranged from 1000 to 10,000 epochs. Moreover, FUZAMP demonstrates better performances than BP based fuzzy neural networks. Following our list presented earlier of desired properties of supervised fuzzy neural networks, Table 3.3 summarizes the comparison of four fuzzy neural networks based on the simulation results provided by Hayashi and Buckley et al. (1993), Ishibuchi et al. (1993, 1994a, 1994b), Keller and Tahani (1992a, 1992b), and Umano et al. (1992), together with results presented here for FUZAMP. Using the hard decision criterion, FUZAMP processes excellent fitting capability for fuzzy training data and is robust to small changes in fuzzy inputs. Using a soft decision criterion, FUZAMP has excellent fitting capability for fuzzy training data and excellent generalization for new fuzzy inputs.

Simulations 5-7: FUZAMP for Classifications

Fuzzy data sets for classification in Simulations 5-7 are similar to those given by Ishibuchi, et. al (1993). Simulation 5 utilized a training set consisting of four two-dimensional fuzzy data pairs, with fuzzy target 0, given as "Class 1", and four two-dimensional fuzzy data pairs, with fuzzy target 1, given as "Class 2":

Class 1: {[(0.15,0.1,0.1),(0.15,0.1,0.1)],
 [(0.15,0.1,0.1),(0.55,0.15,0.15)],
 [(0.45,0.15,0.15),(0.25,0.1,0.1)],
 [(0.5,0.1,0.1),(0.5,0.1,0.1)]}

Class 2: {[(0.3,0.15,0.15),(0.9,0.1,0.1)],
 [(0.6,0.1,0.1),(0.6,0.1,0.1)],
 [(0.7,0.15,0.15),(0.85,0.1,0.1)],
 [(0.8,0.1,0.1),(0.25,0.2,0.2)]}

Table 3.3 Comparison of Four Fuzzy neural Networks

Desired Properties of Fuzzy Neural Networks	Keller's Approach	Umano's Approach	Ishibuchi's Approach	FUZAMP
On-Line Learning	No	No	No	Yes
Stability	Not Guaranteed	Not Guaranteed	Not Guaranteed	Guaranteed
Convergence	Slow	Slow	Slow	Fast
Fitting Capability	Good	Good	Poor	Excellent
Robust	Good	Good	Poor	Excellent
Generalization	No	No	Excellent	Excellent

where (c, δ, γ) is a triangular fuzzy number defined by (2.1). Ishibuchi et. al. (1993) examined the fuzzy classification performance of their BP algorithm with five hidden nodes on this nonlinear classification problem. FUZAMP performance on this problem, with baseline vigilance parameter $\bar{\rho}_a = 0.8$, target vigilance parameter $\rho_b = 1$, fuzzy sensitivity parameter $\tau \rightarrow \infty$, and number of FUZART category nodes equal to 6, is depicted in Fig. 3.15. The rectangles in Fig. 3.15 represent the support (2.43) of two-dimensional fuzzy data. The nonlinear decision boundary was obtained based on a series of fuzzy testing data with spreads equal to zero (i.e., fuzzy singletons). The simulation results match our human intuition and are comparable to those provided by Ishibuchi, et. al. (1993). FUZAMP converges in one epoch, while the BP algorithm (Ishibuchi, et. al., 1993) required 1000 epochs.

The training fuzzy data set in Simulation 6 consists of nine pairs of two-dimensional data with three classes, given by Ishibuchi, et. al. (1993):

Class 1: $\{[(0.2, 0.05, 0.05), (0.85, 0.1, 0.1)],$
 $[(0.3, 0.1, 0.1), (0.55, 0.1, 0.1)],$
 $[(0.5, 0.15, 0.15), (0.8, 0.1, 0.1)]\}$

Class 2: $\{[(0.2, 0.1, 0.1), (0.45, 0.1, 0.1)],$
 $[(0.25, 0.15, 0.15), (0.15, 0.1, 0.1)]$
 $[(0.55, 0.1, 0.1), (0.15, 0.1, 0.1)]\}$

Class 3: $\{[(0.65, 0.1, 0.1), (0.65, 0.15, 0.15)],$
 $[(0.9, 0.05, 0.05), (0.85, 0.1, 0.1)]$
 $[(0.85, 0.15, 0.15), (0.4, 0.05, 0.05)]\}$

where again, (c, δ, γ) is a triangular fuzzy number defined by (2.1). Ishibuchi et. al. (1993) examined the performance of their BP algorithm with five hidden nodes on this multi-class classification problem. FUZAMP was trained with baseline vigilance parameter

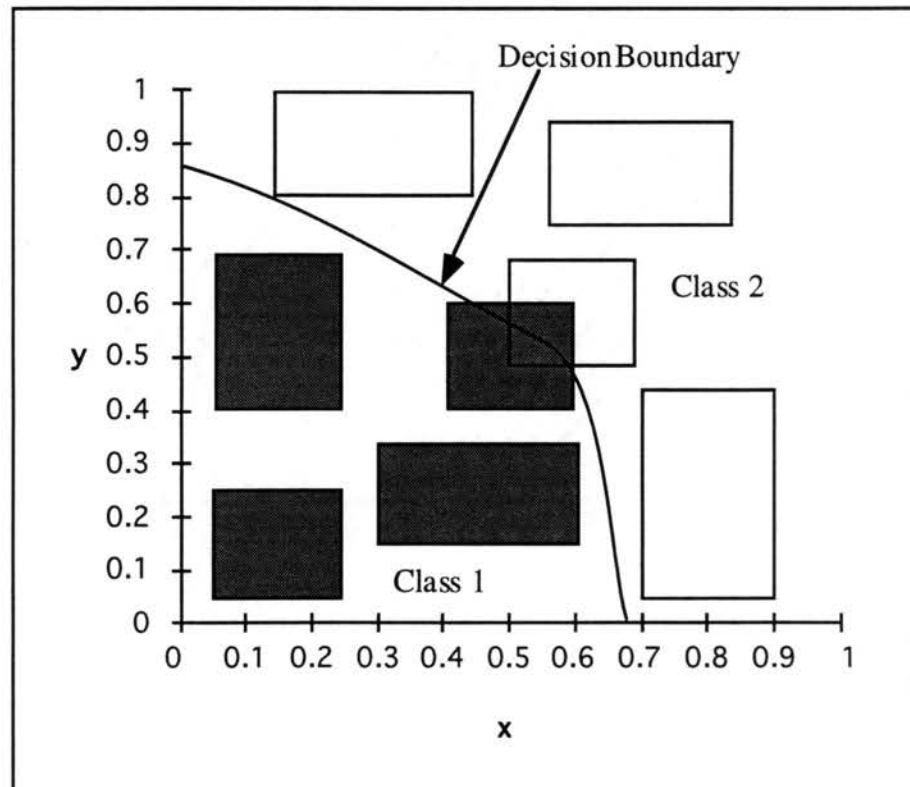


Fig. 3.15 Decision Boundary of a Two-Classifier for Fuzzy Inputs.
 (Shaded and Non shaded Rectangle Represent Fuzzy Support of
 Input From Class 1 and Class 2.)

$\bar{\rho}_a = 0.85$, target vigilance parameter $\rho_b=1$, fuzzy sensitivity parameter $\tau \rightarrow \infty$, and the number of FUZART category nodes equal to 9. Fig. 3.16 shows the simulation results. The decision boundaries were obtained in the same manner as for Simulation 5. Again, FUZAMP achieved the same classification performance as that given by Ishibuchi, et. al. (1993), but required only two training epochs, while the BP algorithm (Ishibuchi, et. al., 1993) required 1000 training epochs.

In Simulation 7, we utilized a training set consisting of ten two-dimensional numerical (i.e., non-fuzzy) data sets (x, y) , with fuzzy targets 0 and 1, respectively, given as "Class 1" and "Class 2" (Ishibuchi et al., 1993):

Class 1: [(0.2, 0.55), (0.4, 0.55), (0.55, 0.15), (0.65, 0.2), (0.65, 0.5)]

Class 2: [(0.1, 0.65), (0.3, 0.7), (0.65, 0.1), (0.7, 0.15), (0.7, 0.7)]

Assume that we also have the following fuzzy rules:

IF X is Small and Y is Small, THEN the Data Set belongs to Class 1

IF X is Large and Y is Medium, THEN the Data Set belongs to Class 2

where "Small", "Medium", and "Large" are defined by the membership functions shown in Fig. 3.17. From the two fuzzy rules, we can add the following fuzzy data to the numerical data set:

Class 1: (Small, Small)

Class 2: (Large, Medium)

FUZAMP performance on a test set consisting of only singleton data using fuzzy complement coding, baseline vigilance parameter $\bar{\rho}_a = 0.5$, target vigilance parameter $\rho_b=1$, fuzzy sensitivity parameter $\tau \rightarrow \infty$, and number of FUZART category nodes equal to 3, is depicted in Fig. 3.18. In contrast, Fig. 3.19 shows FUZAMP performance on a test set consisting of both singleton and fuzzy data using fuzzy complement coding, baseline

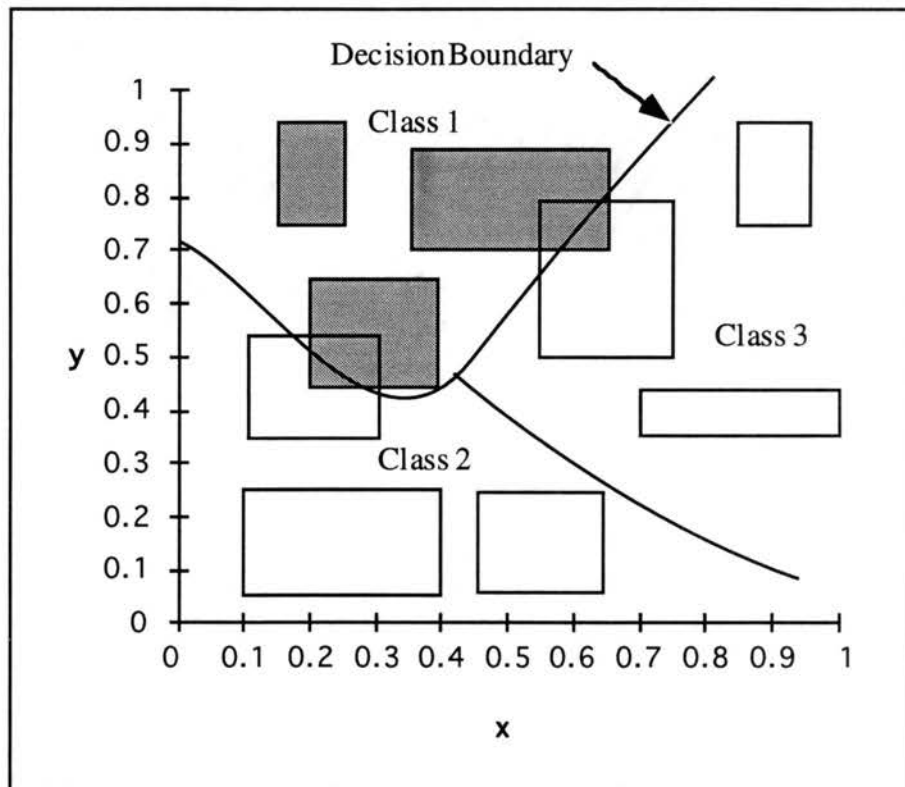


Fig. 3.16 Decision Boundary of a Three-Classifer for Fuzzy Inputs. (Shaded and Non shaded Rectangle Represent Fuzzy Support of Input)

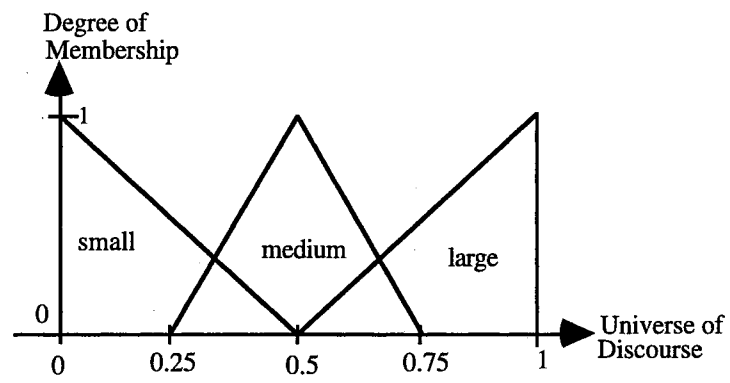


Fig. 3.17 Membership Functions for "small", "medium", and "large" in Simulation 7

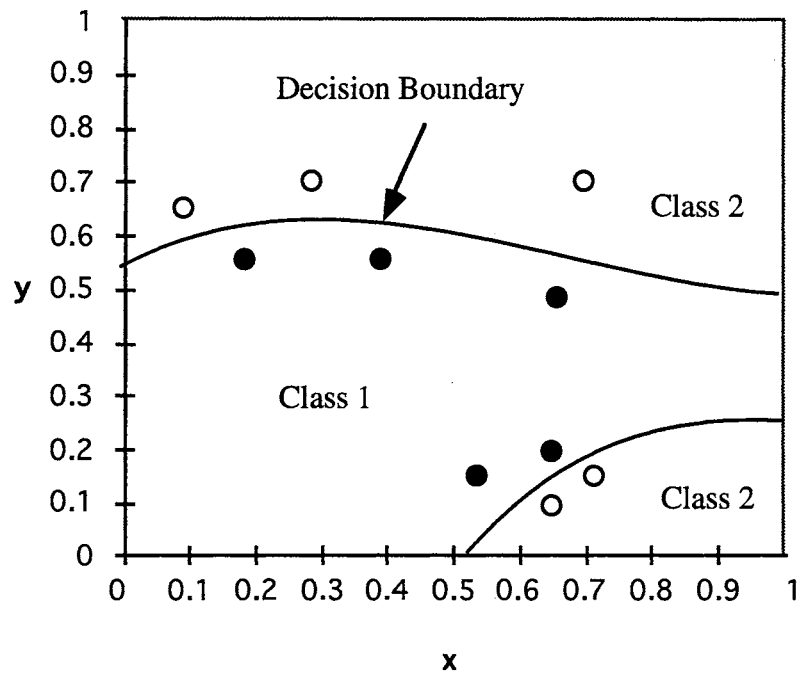


Fig. 3.18 Decision Boundary of a Two-Classifer Using Only Numerical Data In Simulation 7
(Closed and open circles indicate data from Class 1 and Class 2)

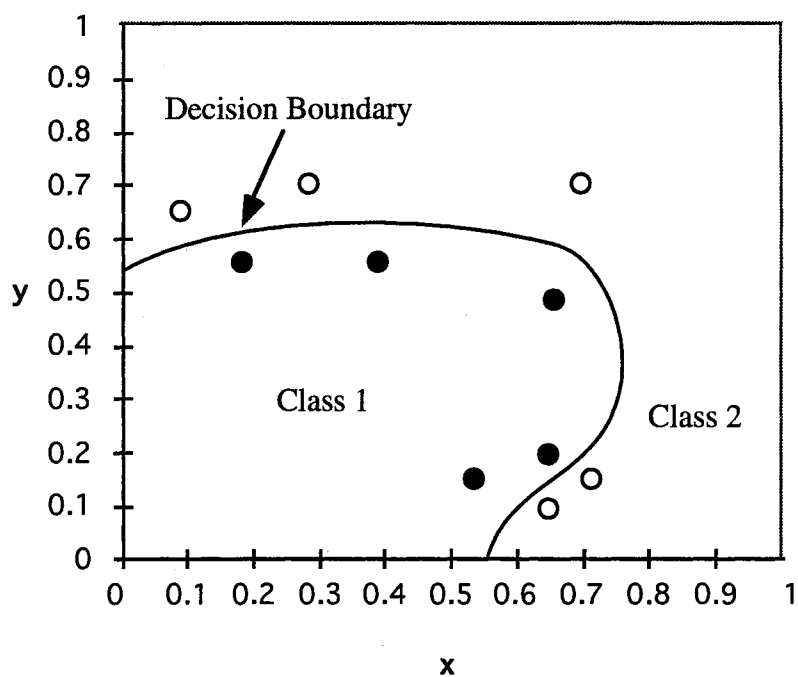


Fig. 3.19 Decision Boundary of a Two-Classifer Using Both Numerical Data and Fuzzy Rules In Simulation 7 (Closed and open circles indicate data from Class 1 and Class 2)

vigilance parameter $\bar{\rho}_a = 0.5$, target vigilance parameter $\rho_b=1$, fuzzy sensitivity parameter $\tau \rightarrow \infty$, and number of FUZART category nodes equal to 4. The classification boundaries determined by FUZAMP from the test sets are shown in solid, curved lines in Figs 3.18 and 3.19. From Fig. 3.18, we note that FUZAMP classifies the given numerical data set correctly. Fig. 3.19 shows that FUZAMP conveys the effects of fuzzy rules very well. FUZAMP converged in one training epoch for this simulation for both Figs 3.18 and 3.19, while the back propagation algorithm (Ishibuchi, et. al., 1993) required 1000 epochs for a similar problem.

Summary

A new fuzzy neural network scheme called FUZAMP has been developed that can quickly and efficiently handle hybrid mixtures of fuzzy data and numerical data. This fuzzy neural network can be applied to classification problems with non-linearly separable fuzzy data, and can also be employed as a fuzzy inference engine using linguistic knowledge described by fuzzy rules. Stability in fuzzy weight learning is guaranteed. Simulation results have shown that FUZAMP has superior fuzzy classification and fuzzy inference capability and stability with fuzzy data. The advantages of FUZAMP compared with other fuzzy neural networks (Hayashi and Buckley et al., 1993; Ishibuchi et al., 1993, 1994a, 1994b; Keller and Tahani, 1992a, 1992b) are that FUZAMP can realize faster and more efficient training for fuzzy data and achieve better performances. Possible applications of FUZAMP to practical problems include fuzzy expert systems, fuzzy control, and fuzzy classification.

CHAPTER IV

FUZZY LOGIC CONTROLLER FOR AUTOMATIC VISION PARAMETER ADJUSTMENT

Introduction

Before applying the new fuzzy neural network in real machine vision applications, we found it necessary to realize automatic adjustment of camera vision parameters "gain" and "offset". This was required to compensate for power fluctuations, changes in ambient light, and camera sensitivity drift in our machine-vision-based robotic dish handling system to insure reliability and accuracy in automated machine vision recognition and inspection. Figure 4.1 illustrates the configuration of the dish handling system. One objective of this research is to realize reliable and fast sorting and inspection of different types of dishes. The system includes an AdeptOne SCARA robot with a five-joint arm, an end effector designed to handle up to six dishes simultaneously, a conveyor belt with a position encoder, four photo-electric sensors, an Adept vision AGS system with three cameras fixed above the conveyor belt and outside the robot workspace, and an Adept MC controller. In the integrated system, indirect illumination is provided by two banks of fluorescence lights, placed at either side of and parallel with the dishrack, below the level of the dishes. The lighting and cameras are enclosed by a "light box", in which the four sides consist of flat white poster board to evenly reflect light, and the top consists of a clear plastic diffuser to diffuse outside light. Dishes to be inspected pass through the open bottom of the light box. Three gray level vision targets are mounted under each camera and above the dishrack for automatic vision parameter adjustment. When a maximum of six identical small dishes or three identical large dishes in a row of the dishrack is presented by the moving conveyor in

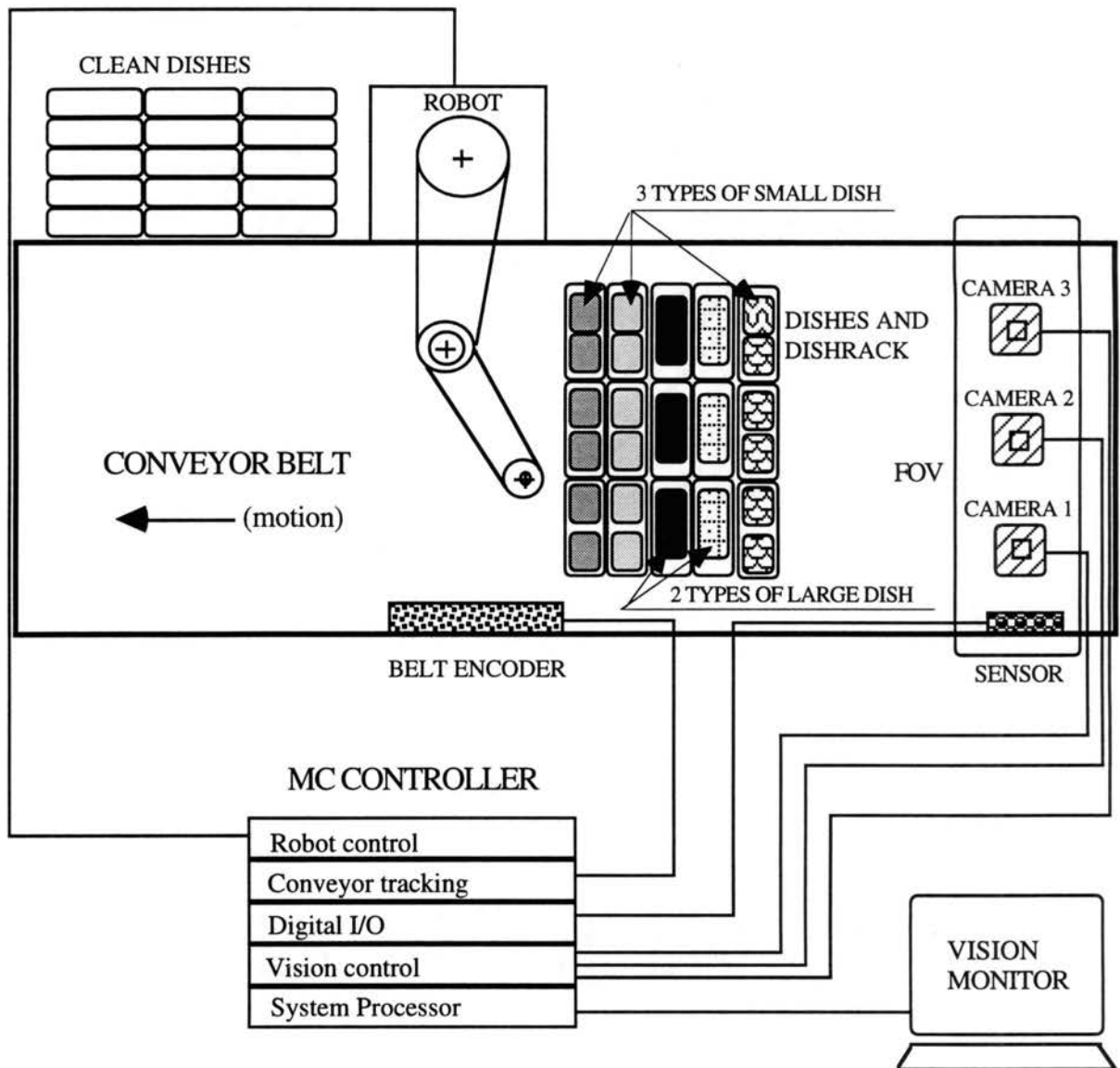


Fig. 4.1 Configuration of a Vision-Based Robotic System

the fields of view (FOV) of the cameras, a photoelectric synchronization sensor is triggered by the dishrack, and the synchronization signal is sent to the MC controller, which directs the three cameras to acquire dish images. The vision task recognizes the type of dishes and inspects the cleanliness of dishes based on the dish image. The results of recognition and inspection are stored in a knowledge database. The robot task continuously monitors the tracking status of dishes on the moving conveyor, retrieves the vision task information from the database, and controls the robot to remove only clean dishes from the dishrack and place them in the proper location off the conveyor. Unclean dishes are left in the dishrack for conveying to a soak tank and rewashing. With no vision parameter control, power fluctuations, changes in ambient light, and camera sensitivity drift affect the reliability and accuracy of automated recognition and inspection of dishes. Because no dynamic model is available to predict camera performance, we considered a fuzzy logic controller to realize automatic adjustment of the camera parameters "gain" and "offset" to compensate for such changes. Not only does fuzzy logic control require no mathematical model of a system, but it also is almost always suitable for a nonlinear MIMO (Multi-Input, Multi-Output) system (Lee, 1990). Moreover, an FLC can translate qualitative and imprecise linguistic statement about control procedures into computer algorithms using "IF-THEN" rules. Decisions through "fuzzy inference" can then be quickly made. Lastly, for our particular control problem, such judgments as "the gain is too large" or "the offset is too small" contain a certain amount of "fuzziness". This appears to be a natural application for FLC.

224
application

Problem Formulation

In our vision-based integrated dish handling system, a stable CCD (Charge-Coupled Device) camera output and consistent lighting are highly desirable. However, power fluctuations, inconsistent ambient lighting and CCD output drift are often unavoidable in practical situations. This causes the different gray levels of a dish to vary.

In order to compensate for these variations, a real time MIMO multi-layer fuzzy controller is proposed for each camera. The controller uses the histogram of a multi-gray level vision target to control the camera parameters "gain" and "offset" to yield desired minimum and maximum gray levels of the target. While the vision task is waiting for the next row of dishes, the camera controller is executed at least once, which provides for near real-time control. Each time the FLC is executed, the vision parameter "threshold" is updated simultaneously. Unlike a conventional PID (Proportional-Integral-Derivative), an FLC can achieve the goals of steady output and satisfactory transient performance simultaneously (Lee, 1990). However, choices of an appropriate set of rules and membership functions with appropriate shape, bandwidth and percentage of overlap significantly affects achieving these goals. Speed and accuracy are two important elements in the design of an FLC. The typically-used triangular membership functions with symmetry and 50% overlap require a small number of rules to express the fuzzy relation, which can reduce both the on-line computational overhead and the memory requirements. With a fixed number of fuzzy subsets, large bandwidths can decrease transient response time, but produces undesirable overshoot or oscillation. Small bandwidths may yield acceptable steady state output but at the same time may increase transient response time. In this work, a multi-layer fuzzy controller with different bandwidths in each layer is used to deal with this problem.

Fuzzy Logic Controller for Adjustment of Gain and Offset

Rule Base

The rule base provides the control goal and the control policy by a set of linguistic rules. A multi-layer rule base for two state variables and two control variables can be represented as:

{ IF X1 and X2 are in Layer j THEN

$$\text{IF } X1=A_{11}^{(j)} \text{ and } X2=A_{21}^{(j)} \text{ THEN } Y1=B_{11}^{(j)} \text{ and } Y2=B_{21}^{(j)} \} \quad (4.1)$$

where $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$; n denotes total rules; m denotes total layers; X_1 and X_2 are fuzzy state variables; $A_{1i}^{(j)}$ and $A_{2i}^{(j)}$ are linguistic values of fuzzy state variables X_1 and X_2 in the universe of discourse $U_1^{(j)}$ and $U_2^{(j)}$; Y_1 and Y_2 are fuzzy control variables; and $B_{1i}^{(j)}$ and $B_{2i}^{(j)}$ are linguistic values of control variables in the universe of discourse $V_1^{(j)}$ and $V_2^{(j)}$. Since both the antecedents and the consequents of these IF-THEN rules contain two linguistic variables, the system is a two-input, two-output fuzzy system. Assume for our vision system that the desired minimum and maximum gray levels of a vision target are x_{rmin} and x_{rmax} , respectively, in the histogram. A variety of external influences, including CCD camera intensity drift and changes in ambient light, can result in changes of minimum and maximum gray levels of the vision target. The task of the control algorithm is to maintain the minimum and maximum gray levels of the vision target at x_{rmin} and x_{rmax} respectively. The FLC is designed to have two fuzzy state variables and two fuzzy control variables. The first state variable e_{min} is the difference between the actual minimum gray level x_{min} of the vision target and x_{rmin} given by:

$$e_{min} = x_{min} - x_{rmin} \quad (4.2)$$

The second fuzzy state variable e_{max} is the difference between x_{rmax} and the actual maximum gray level x_{max} of the vision target, given by:

$$e_{max} = x_{rmax} - x_{max} \quad (4.3)$$

x_{rmin} and x_{rmax} define two areas of a vision target with gray-level less than x_{rmin} and larger than x_{rmax} , respectively, as illustrated in Fig. 4.2. These areas are the allowable maximum amount of spill over the sides of the histogram of the vision target. The two fuzzy control variables are "change of gain" and "change of offset". Table 4.1 and Table 4.2 show the set of rules of a two-layer FLC for gain and offset adjustment, which were developed based on experience with operating the system.

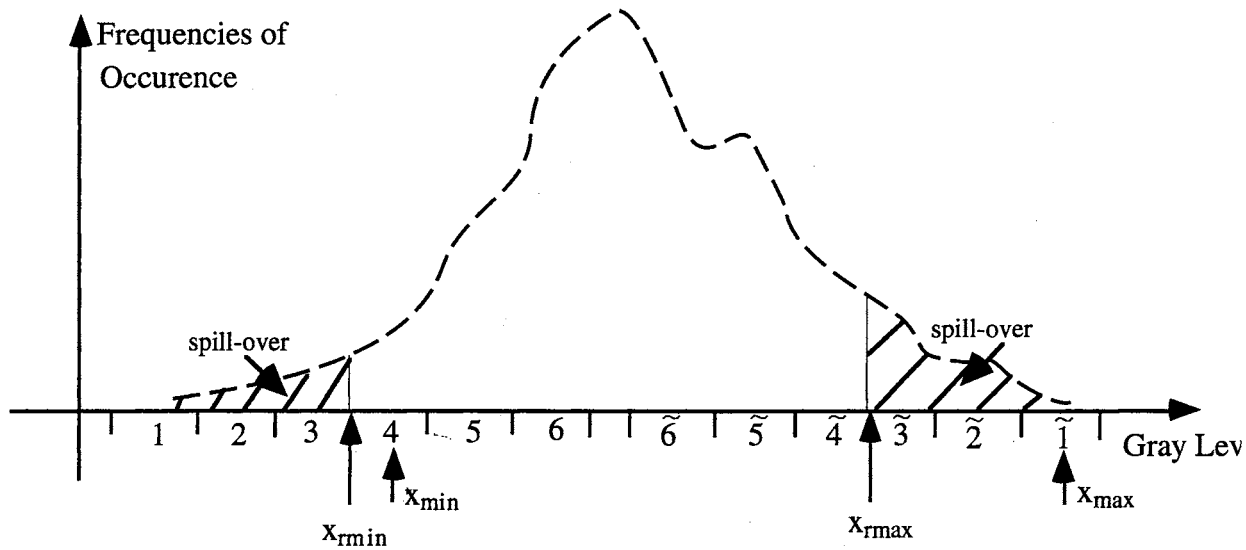


Fig. 4.2 Gray Level Histogram of A Vision Target

Table 4.1 Linguistic Rules of the First Layer of an FLC

$\begin{matrix} e_{min} \\ e_{max} \end{matrix}$	NL	NM	NS	ZE	PS	PM	PL
NL	GNL(OZE)	GNM(OZE)	GNM(OPS)	GNS(OPS)	GNS(OPM)	GZE(OPM)	GZE(OPL)
NM	GNM(ONS)	GNM(OZE)	GNS(OZE)	GNS(OPS)	GZE(OPS)	GZE(OPM)	GPS(OPL)
NS	GNM(ONS)	GNS(ONS)	GNS(OZE)	GNS(OZE)	GZE(OPS)	GZE(OPM)	GPS(OPM)
ZE	GNS(ONM)	GNS(ONS)	GZE(ONS)	**	GPS(OPS)	GPS(OPS)	GPS(OPM)
PS	GNS(ONM)	GZE(ONM)	GZE(ONS)	GZE(ONS)	GPS(OZE)	GPS(OZE)	GPM(OPS)
PM	GZE(ONL)	GZE(ONM)	GZE(ONS)	GPS(ONS)	GPS(OZE)	GPM(OZE)	GPM(OPS)
PL	GZE(ONL)	GZE(ONM)	GPS(ONM)	GPM(ONM)	GPM(ONS)	GPM(OZE)	GPL(OZE)

G=Gain; O=Offset; P=Positive; N=Negative; L=Large; M=Medium; S=Small; ZE=Zero

** : Pass to second layer (Table 2)

Table 4.2 Linguistic Rules of the second Layer of an FLC

$\begin{matrix} e_{min} \\ e_{max} \end{matrix}$	NZ	ZE	PZ
NZ	GZE(OZE)	GNZ(OPZ)	GZE(OPZ)
ZE	GNZ(ONZ)	GZE(OZE)	GPZ(OPZ)
PZ	GZE(ONZ)	GPZ(ONZ)	GPZ(OZE)

NZ=Negative Zero; ZE=Zero; PZ=Positive Zero; G=Gain; O=Offset

As an example in interpreting Table 4.1, consider the entries in the 3rd column, 6th row. This should be read, "IF e_{\min} is negative small (NS) and e_{\max} is positive medium (PM), the change of gain should be set to zero (ZE) and the change of offset set to negative small (NS)." As shown in Fig. 4.2, the desired minimum and maximum gray-levels are $x_{r\min}$ and $x_{r\max}$, respectively. When the actual minimum gray-level x_{\min} of the vision target is in the region 4-6, and the actual maximum gray-level x_{\max} in the region $\tilde{4}\text{-}\tilde{6}$ respectively, the vision parameter "gain" should be increased, which would expand the gray-level range of the vision target. Usually, x_{\min} and x_{\max} lying in different regions require different adjustments of the gain and offset. Some regions of x_{\min} and x_{\max} may require a large increase (decrease) of gain and offset, and some may require a medium or small increase (decrease) of gain and offset. Table 4.1 and Table 4.2 show all regions of x_{\min} and x_{\max} and their corresponding control actions. Note that if both e_{\min} and e_{\max} are ZERO in the first layer (Table 4.1), the control action is passed to the second layer (Table 4.2), which is smoothed naturally by the interpolative capability of the fuzzy logic. For the control problem described herein, the second layer of the FLC partitions the ZERO fuzzy set of the first layer into three subsets, which would result in a tendency toward smaller control outputs so that the system would be damped, reducing the oscillations of the system output.

Membership Functions

Membership functions are used to transform crisp inputs into fuzzy sets in the process of fuzzification and fuzzy sets back into crisp outputs in the process of defuzzification. While there are many types of membership functions, in this work isosceles triangles are chosen to simplify FLC computation. A 50% overlap in the membership functions of all the fuzzy sets is used so that only two fuzzy sets have non-zero degree of membership at any given point of the universe of discourse, which would reduce on-line computational overhead. The universe of discourse of the two controller

inputs e_{\min} and e_{\max} is chosen to cover the entire range of possible input values, which makes input scaling unnecessary. The universe of discourse of inputs and outputs is partitioned into seven membership functions corresponding to seven linguistic variables (VS, NM, NS, ZE, PS, PM, PL) in the first layer of the FLC as illustrated in Fig. 4.3(a). Similarly, The universe of discourse of inputs and outputs in the second layer of the FLC are partitioned into three membership functions corresponding to three linguistic variables (NZ, ZE, PZ) as illustrated in Fig. 4.3(b). The two controller inputs e_{\min} and e_{\max} are converted into fuzzy singletons by fuzzification. Basically, fuzzy singletons define controller inputs as fuzzy sets with the degree of membership equal to one at e_{\min} and e_{\max} and zero for all other points in the universe of discourse.

As shown in Fig. 4.3(a) and Fig. 4.3(b), only four rules in the rule base are valid for any fuzzy input singleton. For a specific e (e denotes e_{\min} or e_{\max}) in the universe of discourse, the degree of membership $\mu_p^{(L)}(e)$ for the leftmost fuzzy set in the first layer of the FLC (Fig. 4.3(a)) can be represented as:

$$\mu_p^{(L)}(e) = \begin{cases} 1 & p = 0 \text{ or } 7 \\ (B(p-3)-e)/B & \text{otherwise} \end{cases} \quad (4.4)$$

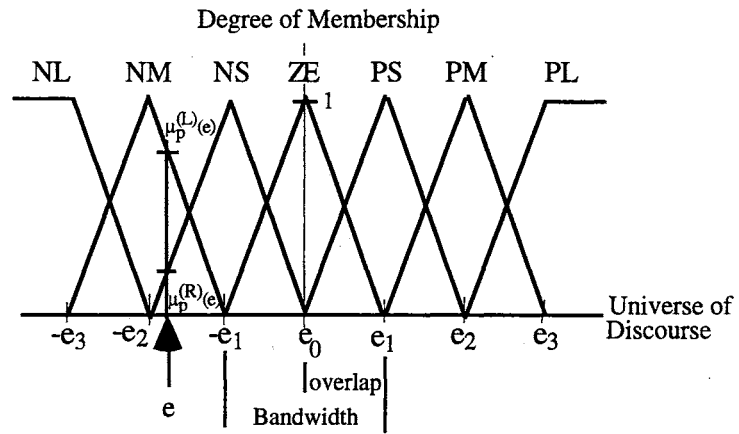
$$p = \min[7, \max(0, \text{INT}((e+4B)/B))] \quad (4.5)$$

where L denotes leftmost; B is half of the bandwidth of the triangular membership functions; INT denotes integer. The degree of membership for rightmost fuzzy set, $\mu_p^{(R)}(e)$, is the complement of that for the leftmost, namely, $\mu_p^{(R)}(e) = 1 - \mu_p^{(L)}(e)$. Similarly, the degree of membership for the leftmost fuzzy set in the second layer of the FLC (Fig. 4.3(b)) can also be represented as:

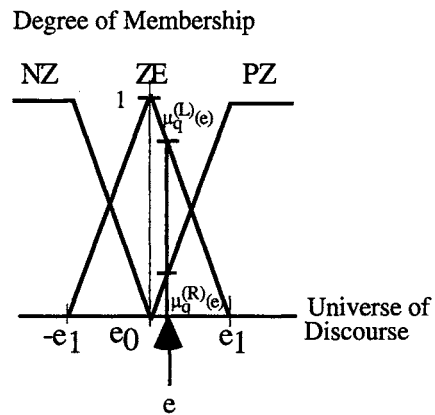
$$\mu_q^{(L)}(e) = \begin{cases} 1 & p = 0 \text{ or } 3 \\ (B(q-2)-e)/B & \text{otherwise} \end{cases} \quad (4.6)$$

$$q = \min[3, \max(0, \text{INT}((e+3B)/B))] \quad (4.7)$$

The degree of membership for rightmost fuzzy set, $\mu_q^{(R)}(e)$, is the complement of that for the leftmost, namely, $\mu_q^{(R)}(e) = 1 - \mu_q^{(L)}(e)$.



(a) Fuzzy Sets for the First Layer of FLC



(b) Fuzzy Sets for the Second Layer of FLC

Fig. 4.3

Inference and Defuzzification

Based on the fuzzy input singletons, an inference process is initiated. For a specific e_{\min} and e_{\max} , the rule base is executed and the most appropriate rules activated (Lee, 1990). Each rule in Table 4.1 and Table 4.2 defines a MIMO fuzzy implication which can be represented as two MISO fuzzy implications given by:

$$\mu_{R_{1i}^{(j)}}(X1, X2, Y1) = \mu_{A_{1i}^{(j)} \times A_{2i}^{(j)}}(X1, X2) \rightarrow \mu_{B_{1i}^{(j)}}(Y1) \quad (4.8)$$

$$\mu_{R_{2i}^{(j)}}(X1, X2, Y2) = \mu_{A_{1i}^{(j)} \times A_{2i}^{(j)}}(X1, X2) \rightarrow \mu_{B_{2i}^{(j)}}(Y2) \quad (4.9)$$

where μ denotes degree of membership; i denotes i th rule; j denotes j th layer; $X1$ and $X2$ denote fuzzy state variables e_{\min} and e_{\max} , respectively; $Y1$ and $Y2$ denote fuzzy control variables "change of gain" and "change of offset", respectively; $A_{1i}^{(j)} \times A_{2i}^{(j)}$ is a fuzzy set in $U1^{(j)} \times U2^{(j)}$; $R_{1i}^{(j)}$ and $R_{2i}^{(j)}$ are fuzzy implications in $U1^{(j)} \times U2^{(j)} \times V1^{(j)}$ and $U1^{(j)} \times U2^{(j)} \times V2^{(j)}$ respectively, and \rightarrow denotes a fuzzy implication function. Using the product operation rule of fuzzy implication (Lee, 1990), we have:

$$\mu_{R_{1i}^{(j)}}(X1, X2, Y1) = \mu_{A_{1i}^{(j)} \times A_{2i}^{(j)}}(X1, X2) \cdot \mu_{B_{1i}^{(j)}}(Y1) \quad (4.10)$$

$$\mu_{R_{2i}^{(j)}}(X1, X2, Y2) = \mu_{A_{1i}^{(j)} \times A_{2i}^{(j)}}(X1, X2) \cdot \mu_{B_{2i}^{(j)}}(Y2) \quad (4.11)$$

Based on the rule base, the fuzzy inference engine employs the compositional rule of inference to infer its output fuzzy set. Assume that $A_{1x}^{(j)} \times A_{2x}^{(j)}$ is a fuzzy set in $U1^{(j)} \times U2^{(j)}$ from the output of Fuzzification. Using the Max-Min Compositional Rule of Inference, the output fuzzy sets from the fuzzy inference engine for each rule is:

$$\mu_{A_{1x}^{(j)} \times A_{2x}^{(j)}} \circ R_{1i}^{(j)} = \max_{(x1, x2) \in U1^{(j)} \times U2^{(j)}} [\min \mu_{A_{1i}^{(j)} \times A_{2i}^{(j)}}(X1, X2), \mu_{R_{1i}^{(j)}}(X1, X2, Y1)] \quad (4.12)$$

$$\mu_{A_{1x}^{(j)} \times A_{2x}^{(j)}} \circ R_{2i}^{(j)} = \max_{(x1, x2) \in U1^{(j)} \times U2^{(j)}} [\min \mu_{A_{1i}^{(j)} \times A_{2i}^{(j)}}(X1, X2), \mu_{R_{2i}^{(j)}}(X1, X2, Y2)] \quad (4.13)$$

Where \circ denotes the Max-Min composition operator.

The fuzzy inference engine delivers its outputs as fuzzy sets. Defuzzification maps from the fuzzy sets into non-fuzzy control outputs that best represent the possible distributions of inferred fuzzy control actions. While there are many strategies for defuzzification (Hellendoorn and Thomas, 1993), the Center-of-Sums (COS) method is used in this work because of its good performance and speed of computation. The control outputs, $y1$ =change of gain and $y2$ =change of offset, corresponding to the COS defuzzification method are:

$$y1 = \sum_{k=1}^m y_{1k}^{(j)} \sum_{i=1}^n \mu_{A_{1x} \times A_{2x}}^{(j)} \circ \mu_{R_{11}}^{(j)}(y_{1k}^{(j)}) / \sum_{k=1}^m \sum_{i=1}^n \mu_{A_{1x} \times A_{2x}}^{(j)} \circ \mu_{R_{11}}^{(j)}(y_{1k}^{(j)}) \quad (4.14)$$

$$y2 = \sum_{k=1}^m y_{2k}^{(j)} \sum_{i=1}^n \mu_{A_{1x} \times A_{2x}}^{(j)} \circ \mu_{R_{21}}^{(j)}(y_{2k}^{(j)}) / \sum_{k=1}^m \sum_{i=1}^n \mu_{A_{1x} \times A_{2x}}^{(j)} \circ \mu_{R_{21}}^{(j)}(y_{2k}^{(j)}) \quad (4.15)$$

where $y_{1k}^{(j)}$ is the discrete point in the universe of discourse $v1^{(j)}$; $y_{2k}^{(j)}$ is the discrete point in the universe of discourse $v2^{(j)}$; m is number of the discrete points; n denotes number of rules. For the membership functions described in this work, $n=4$.

Threshold Control

The vision parameter "threshold" is used to define a binary image (either black or white). Given a gray-level dish image $G(x,y)$, its binary image $B(x,y)$ is defined by thresholding according to:

$$B(x,y) = \begin{cases} L_{\max} & G(x,y) \geq T \\ L_{\min} & \text{otherwise} \end{cases} \quad (4.16)$$

where $L_{\max}=127$ (white), $L_{\min}=0$ (black), and T is a threshold. The two binary levels represent the objects (dishes and food particles with higher gray-levels) and the background (conveyor, dishrack, food particles with lower gray-levels, and dish imperfections). Binary images usually take less time to acquire and process than gray-level images. For the application in our dish handling operation, binary thresholding is proposed for dish image segmentation. The segmented binary dish image is proposed for recognition and sorting of dishes, and both the binary and gray-level images are proposed for inspection. Since

dishes with different colors require different threshold values for segmentation and inspection, appropriate selection of the threshold for each camera significantly affects the system performance. When an image in the FOV consists of only one type of dish and background, a possible approach to choose a threshold is to search a bimodal gray-level histogram and find a gray-level separating the two peaks. However, when more than one type of dish with different colors appears in the FOV, the histogram may no longer be bimodal. In this work, a specific thresholding technique was investigated for our real time system. First, gain and offset are initially adjusted by the FLC so that the minimum and maximum gray-level are set at desired values in the histogram of the vision target. Thresholds for each camera are then chosen off-line for satisfactory image segmentation of all five types of clean dishes by trial-and-error based on the gray-level histogram. Lastly, we set the chosen thresholds and take a white pixel count of the binary image of the vision target (a pixel is the smallest elemental area discernable by the camera/vision system). This white pixel count is taken to be the desired value. In other words, satisfactory dish image segmentation requires that the binary image of the vision target has this desired pixel count, because this pixel count corresponds to the chosen threshold. Each time the FLC is executed, the vision parameter "threshold" for segmentation is updated so that the pixel count of the binary image of the vision target remains unchanged. In similar fashion, the thresholds for inspection of different types of dishes with different colors are simultaneously adjusted, automatically, to provide maximum food particle detection. While there are many algorithms existing for automatically selecting a threshold (Sahoo, et al., 1988), most of them are either ineffective for our application or require excessive time. The technique proposed here proved to be suitable for both image segmentation and dish inspection in our integrated dish handling system.

Experimental Study

The purpose of the FLC proposed in this work is to adapt the vision system to the changing environment for improving overall performance of dish recognition and inspection. In order to evaluate the control strategies described above, experimental studies were made. As illustrated in Figure 4.1, three cameras are fixed above the conveyor belt: Camera 1 (Pulnix Model No.TM-540 Serial No.015207), Camera 2 (Pulnix Model No.TM-540 Serial No.022177), and Camera 3 (Pulnix Model No.TM-540 Serial No.014969), each with a 25 mm focal length lens. Although the three cameras are same make and model, they nevertheless exhibit substantially different static and dynamic behavior because CCD elements in each camera vary due to manufacturing variations. Considering practical adjustable ranges of gain and offset for each camera and to provide maximum food particle detection, the desired minimum and maximum gray-levels of the vision targets for the three cameras were set by trial and error at: $x_{rmin(1)}=10$, $x_{rmax(1)}=60$; $x_{rmin(2)}=10$, $x_{rmax(2)}=80$; $x_{rmin(3)}=10$, and $x_{rmax(3)}=80$, where the numbers in parentheses designate the camera number. The bandwidth of the first layer membership functions of the multi-layer FLC was selected as 6. The bandwidth of the second layer membership functions was chosen as 1.5. For comparison, two single layer FLCs (first layer) with bandwidths of 6 and 1.5, respectively, for membership functions were also evaluated under identical conditions. The experiments were done after the system was started up with original vision parameter settings, namely gain(1)=230, offset(1)=80, gain(2)=210, offset(2)=55, gain(3)=230, and offset(3)=150, where the numbers in parentheses designate the camera number. Figure 4.4 show the minimum gray-level responses of Camera 2. Figure 4.5 show its maximum gray-level response. The response of a single layer FLC with the bandwidth of 6 for membership functions exhibits a large oscillation around the set point when e_{min} and e_{max} are approaching zero. For a single layer FLC with the bandwidth of 1.5 for membership functions, the transient response time increases by a

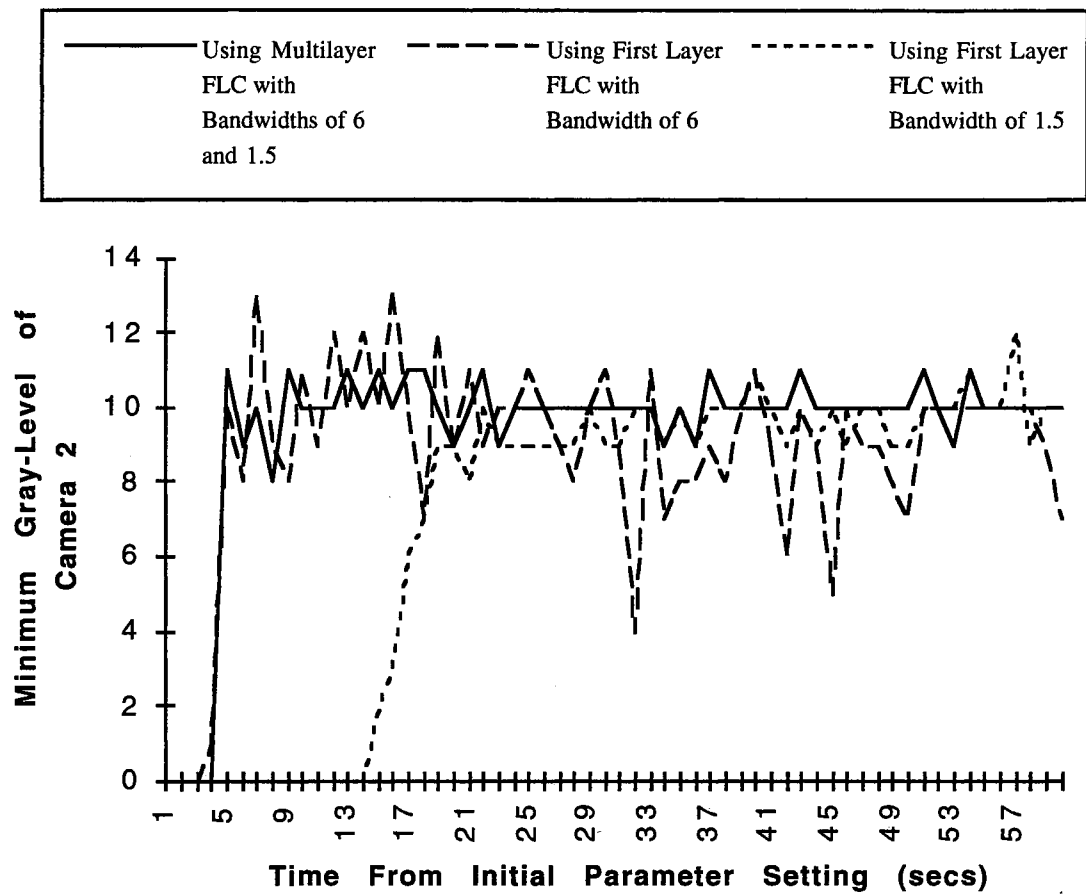


Fig. 4.4 Minimum Gray-Level Response of Camera 2

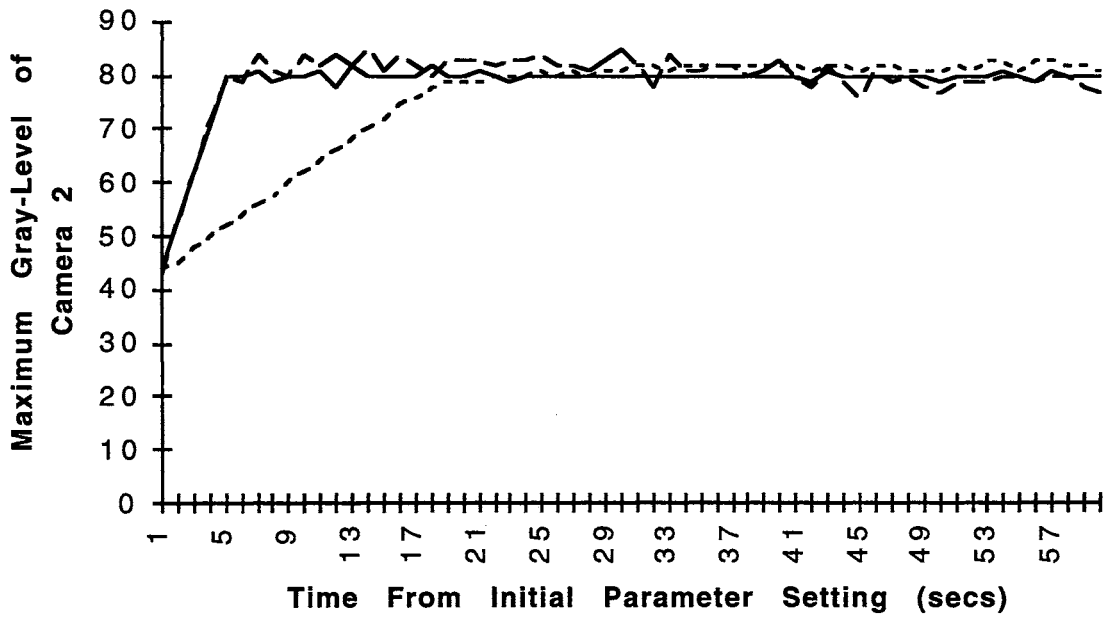
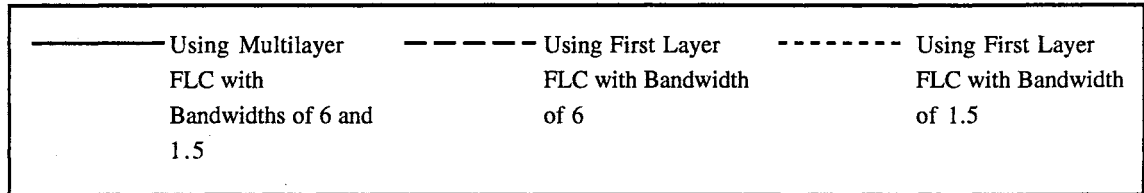


Fig. 4.5 Maximum Gray-Level Response of Camera 2

substantial margin, while the oscillation behavior around a final value is improved. By comparing the two-layer FLC with the two single layer FLCs, one can see that the two-layer FLC performs much better in that it has improved transient speed, overshoot, and accuracy.

To further evaluate overall performance of the implemented two-layer FLC, the minimum and maximum gray-level responses of each camera with and without FLC were observed over a period of 6 hours. Figures 4.6 and 4.7 show the response of each camera with constant gain and offset, 216 and 113 for Camera I, 210 and 96 for Camera II, and 176 and 54 for Camera III, which were initially set by the FLC to provide the desired minimum and maximum gray-level of each camera. As expected, camera output exhibits obvious changes in both minimum and maximum gray-level of the vision target due to coupled effects of power fluctuations, inconsistent ambient lighting and CCD camera output. Figures 4.8 and 4.9 show the minimum and maximum gray-level response of each camera with a two-layer FLC with the bandwidths of 6 and 1.5. The responses are much more stable than those in Fig. 4.6 and 4.7.

Summary

A MIMO multi-layer fuzzy logic controller for real time adjustment of vision parameters has been proposed in this work. Since many processes are controlled by PID controllers, the question may arise as to whether the FLC offers an improvement over conventional PID controllers. First, conventional PID controller design requires a model-based approach when the nonlinearity is taken into account. The coupled effects of a MIMO PID controller must often be analyzed mathematically in the design process. FLC is very flexible in handling nonlinear coupled relationships. Second, in the design of a conventional PID controller, trade-offs are usually required between decreasing the steady state error and improving transient performance. FLC can achieve both the goals of good

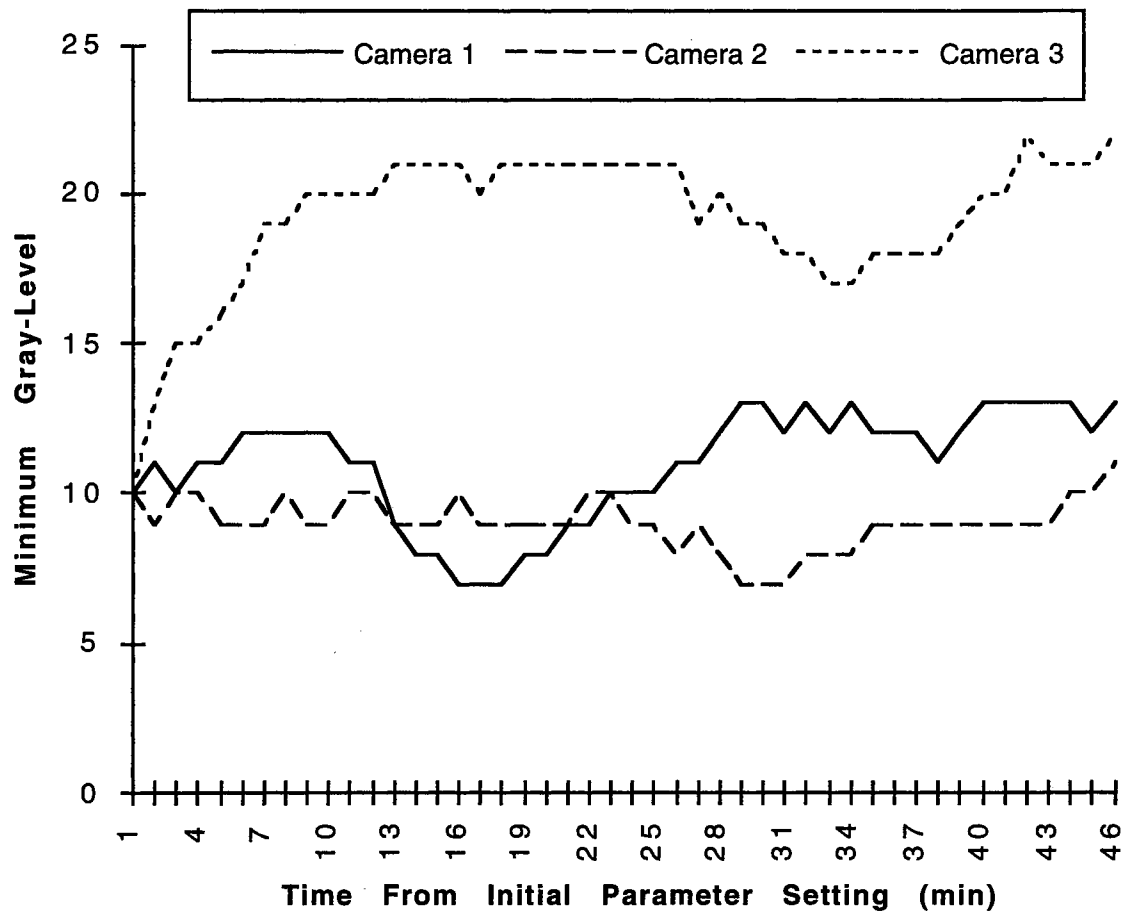


Fig. 4.6 Minimum Gray-Level Response Without FLC

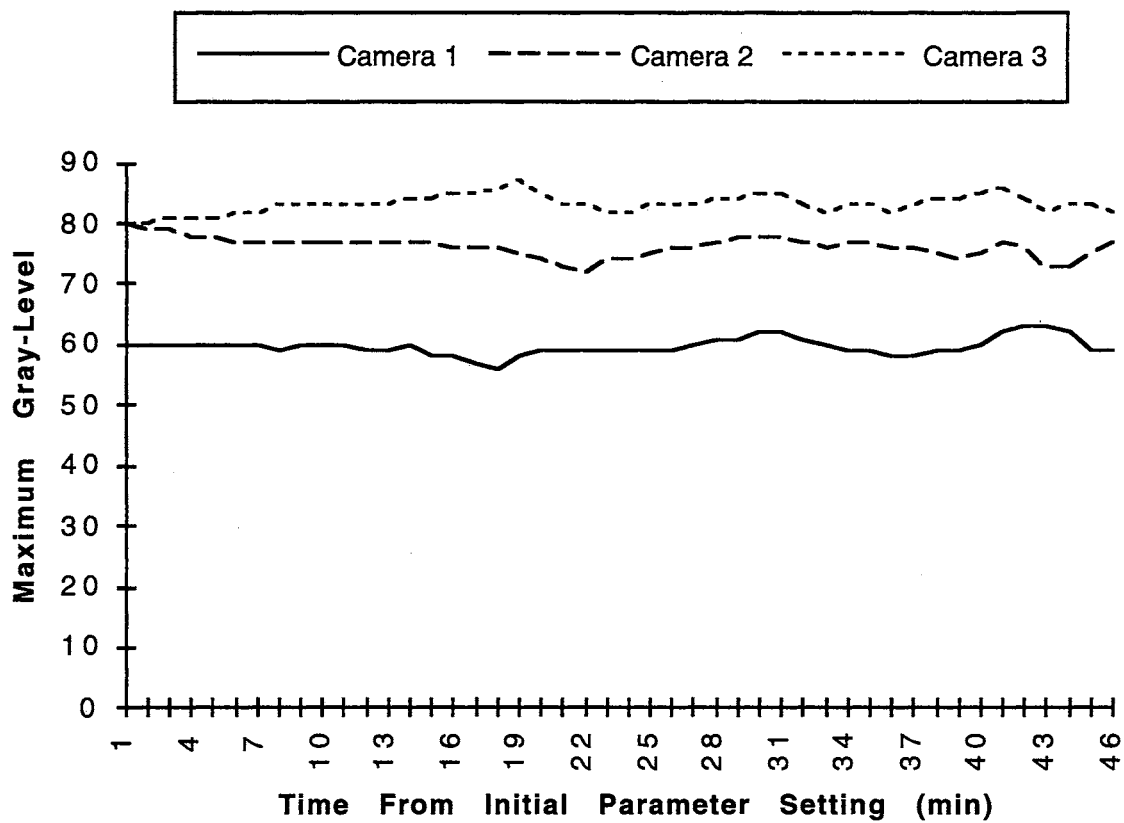


Fig. 4.7 Maximum Gray-Level Response Without FLC

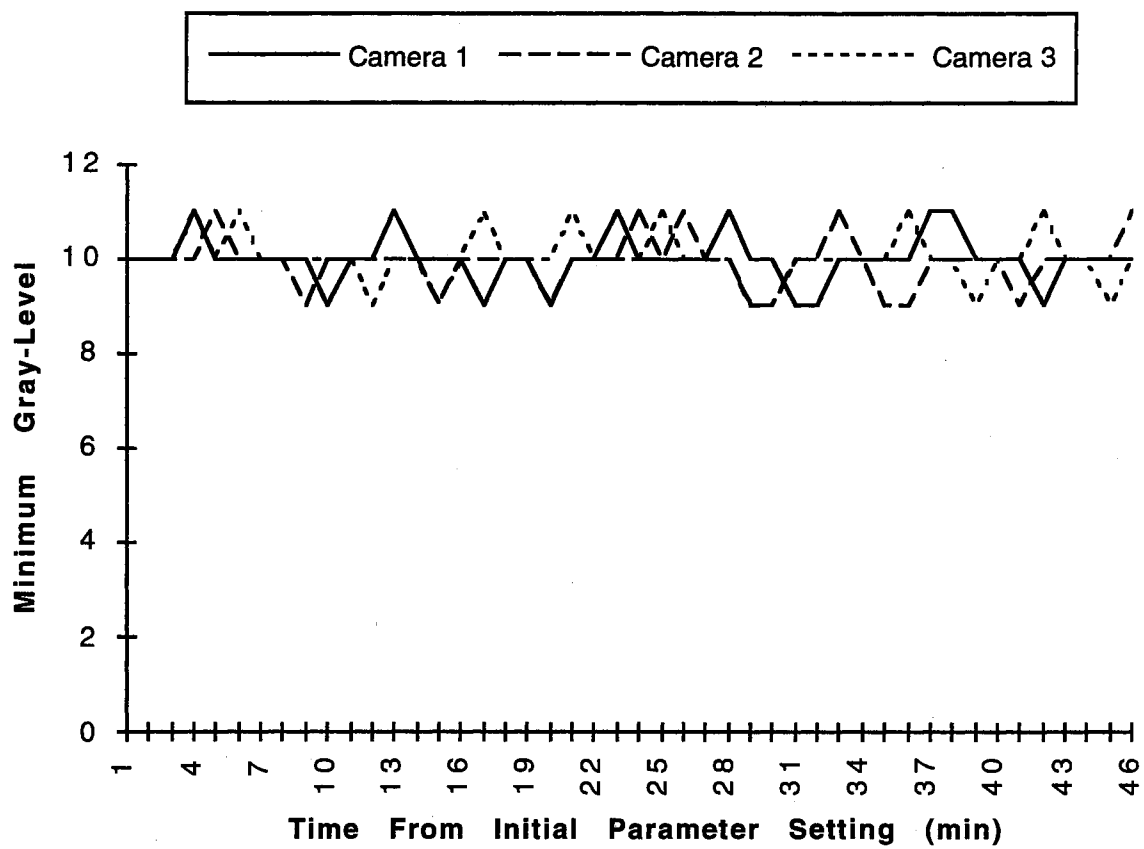


Fig. 4.8 Minimum Gray-Level Response with an FLC

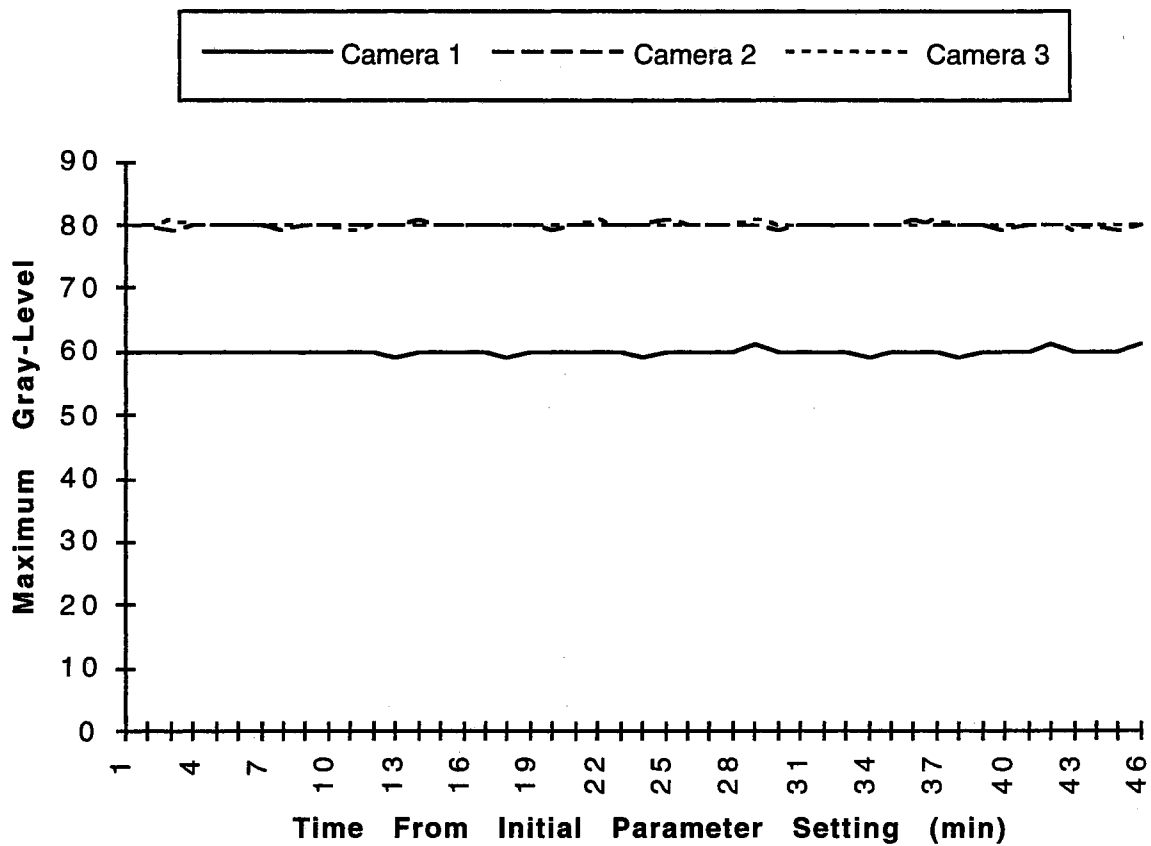


Fig. 4.9 Maximum Gray-Level Response with an FLC

steady output and satisfactory transient performance simultaneously. Since there is no mathematical model available for our vision system, while there is considerable operating experience to provide qualitative statements for fuzzy sentences, the choice of FLC seems most appropriate for our application. In fact, it is shown through experimental studies that the multi-layer FLC performs well in controlling vision parameters. The performance of an FLC can be enhanced significantly by using a multi-layer FLC, yielding faster response with less overshoot than a conventional single layer FLC can achieve. The triangular membership functions with symmetry and 50% overlap result in only four valid rules in the rule base being operative for a given input, which can reduce on-line computational overhead of a multi-layer FLC. The simple algorithm used in this work to calculate the degree of membership for multi-layer FLC helps to increase computing speed. While more fuzzy state variables, such as change of e_{\min} (error of minimum gray-level) and change of e_{\max} (error of maximum gray-level), can be added to the FLC to improve the performance of a multi-layer FLC, the complexity of its implementation and computational overhead may increase significantly for a MIMO system. The fuzzy state variables used in this work are suitable for our control problem.

CHAPTER V

MACHINE VISION FUZZY OBJECT RECOGNITION AND DISH CLEANLINESS INSPECTION USING FUZAMP

Introduction

In machine vision applications, there exist many types of uncertainty such as noise of various sorts, vagueness of representation, and non-random uncertainties in features extracted from camera data. In this chapter, we illustrate dealing with such uncertainties by presenting applications of FUZAMP in machine vision sorting and inspection of dish pieces in a prototype for large commercial dish washing operations. Current large commercial dish washing operations handling 20,000 or more silverware pieces and dishes per day are performed manually in a hot, humid environment. This leads to high operating costs and labor turnover. Automation techniques are sought to perform most of the tasks currently done manually. Developing a high speed, efficient system integrating the sorting and inspection of dishes and silverware pieces into a single working unit is a very challenging task. We first present an application of FUZAMP in machine vision tasks for fast and accurate recognition of highly specular objects in a lighted field, such as polished metal, ceramic, glass, and plastic objects. Highly specular objects with components lying out of the camera image plane typically yield different measured features (such as area and perimeter) for the same object appearing in different positions and orientations in the camera Field of View (FOV). This is because even with carefully designed lighting, different orientations of a three dimensional specular part produce different gray-level images measured by the camera. This yields uncertainty in the visual representation of the object, such that we refer to such objects as "fuzzy objects". For example, typical stainless

steel soup spoons when presented at different locations and orientations in the FOV, including both right-side-up and right-side-down presentation, as illustrated in Fig. 5.1-5.2, yielded a range of normalized area between 0.52 and 0.63 (21% variation), and normalized perimeter between 0.46 and 0.59 (28% variation). The variation was caused by shadows on the spoons of different sizes and locations, which caused the vision system to interpret the shadowed areas as background, rather than parts of the spoons. In this work, a scheme is proposed that uses uncertain information provided by image processing of a machine vision system for such fuzzy objects. The data with uncertainty were transformed into fuzzy sets, and then fuzzy signals were fed to FUZAMP in order to recognize fuzzy objects.

In addition to applying FUZAMP in fuzzy object recognition, we also used FUZAMP to correlate human evaluations with machine evaluations of the cleanliness of dishes. Dish cleanliness inspection is a challenging task for machine vision because food particles on a dish may have many variations in their gray level images, even with small changes in viewing position and direction. Sensor output from a camera typically reflects this uncertainty, and descriptions of "clean" and "dirty" are ill-defined in a mathematical sense. In this work, FUZAMP was implemented to automatically classify a dish as "clean" or "dirty". The classification performance for various real conditions was tested and compared with that using fuzzy ARTMAP proposed by Carpenter and Grossberg et al. (1991a).

The scheme proposed in this chapter for machine vision recognition of fuzzy objects and inspection of dish cleanliness employs combining fuzzy neural networks with machine vision. The data from image processing of a machine vision system includes uncertainty to some extent, which could be transformed into fuzzy sets and used as input to FUZAMP. The transformation, or fuzzification, implies that a given measurement value from the machine vision system is most likely to be the correct value in comparison with all its immediate neighboring values. Nevertheless, neighboring values are also possibly

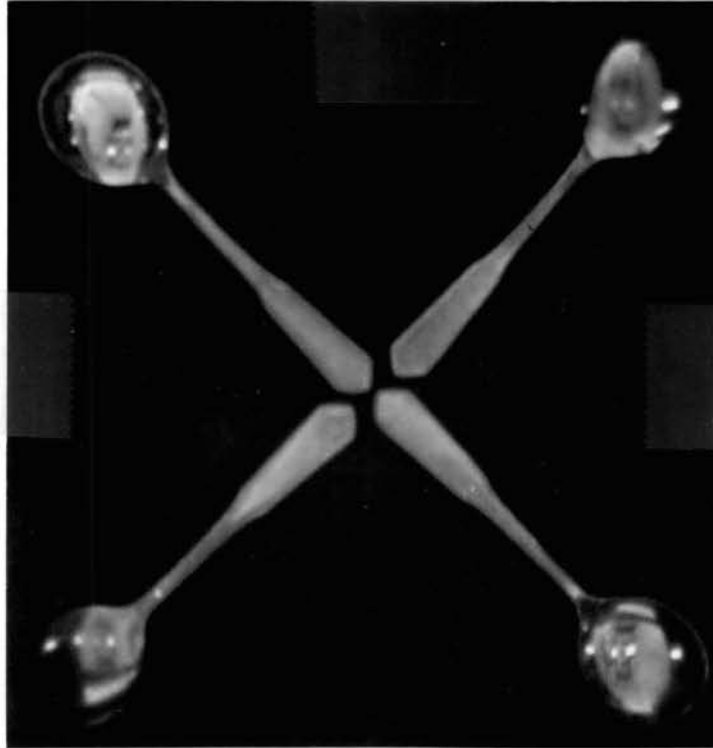


Fig. 5.1 Gray Level Images of Stainless Steel Soup Spoons in the FOV

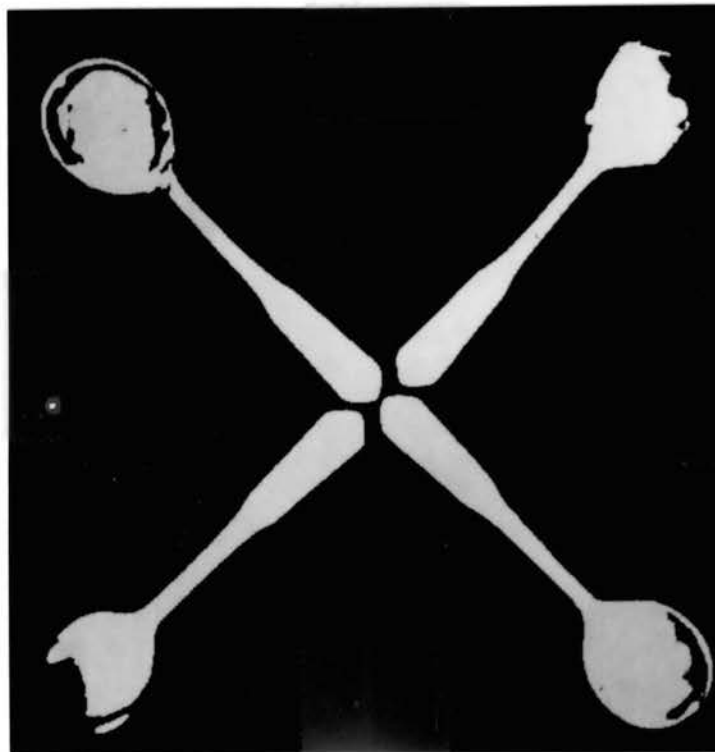


Fig. 5.2 Binary Images of Stainless Steel Soup Spoons in the FOV

correct, but to a lesser degree. Assume that the probability distribution function for the machine vision signal with uncertainty is symmetric. It is natural for the membership functions of fuzzy sets to also be chosen symmetric about a measurement value. Symmetric functions such as triangular, trapezoidal, and Gaussian functions are all appropriate candidates for membership functions of fuzzy sets, but in this paper, fuzzy sets are represented by triangular membership functions. In such functions, a larger "spread" implies that more uncertainty exists in a given measurement.

In what follows, we first applied FUZAMP in fuzzy object recognition of ~~silverware~~. Then we used FUZAMP for dish cleanliness inspection. We employ both FUZAMP and the so-called fuzzy ARTMAP (Carpenter and Grossberg, et al., 1991) neural networks for recognition and inspection problems, using the same training and testing sets and the same features for each network.

Fuzzy Object Recognition Using FUZAMP

Feature Extraction

In our silverware recognition problem, fuzzy patterns used to train FUZAMP were obtained from an Adept Technology, Inc. AGS Machine Vision System interfaced with a CCD camera (Pulnix Model No. TM-540 Serial No. 022177). Features were extracted based on the following sequential procedures: 1. start the vision system and let it warm-up for half an hour; 2. employ fuzzy logic controllers to set the vision parameters "gain", "offset", and "threshold", as described in Chapter IV; 3. present fuzzy objects in the FOV at different locations and orientations in the order of butter knife, large (dinner) fork, small (salad) fork, large (soup) spoon, and small (tea) spoon, respectively, repeating this order until the desired number of training or testing exemplars is obtained; 4. for each presentation, employ the AGS machine vision image processing software to obtain features including the area, the perimeter, and the major radius of the enveloping ellipse of each individual fuzzy object, and store these results in memory; 5. find the maximum of each

feature over the stored set of all exemplars, normalize each feature as in (5.1)-(5.3) below, and restore all normalized data; 6. employ the stored, normalized data in training and testing FUZAMP, as described below. Figs 5.1 and 5.2 show gray level and binary camera images of soup spoons, respectively, while Figs 5.3 and 5.4 show photographs of the remaining silverware pieces. Each feature from the vision system was normalized as follows:

For area,

$$A_n = A/A_{ref} \quad 0 < A_n \leq 1 \quad (5.1)$$

where A_n is the normalized area of the object image, A is the measured area of the object image and A_{ref} is reference area, taken as the maximum area over all images of all silverware pieces. For perimeter,

$$P_n = P/P_{ref} \quad 0 < P_n \leq 1 \quad (5.2)$$

where P_n is the normalized perimeter of the object image, P is the measured perimeter of the object image and P_{ref} is reference perimeter, taken as the maximum perimeter over all images of all silverware pieces. For major radius,

$$R_n = R/R_{ref} \quad 0 < R_n \leq 1 \quad (5.3)$$

where R_n is the normalized major radius of the ellipse enveloping the object image, R is the measured major radius of the ellipse enveloping the object image and R_{ref} is the reference major ellipse radius, taken as the maximum major ellipse radius over all images of all silverware pieces. These normalized features were then fuzzified by appropriate choices of the spreads of triangular membership functions. Appropriate choices of spreads are discussed in the following section.

Training of FUZAMP

In the training process for FUZAMP we have a collection of fuzzy input and output pairs of features, designated as $(\mathbf{p}^1, \mathbf{t}^1)$, $(\mathbf{p}^2, \mathbf{t}^2)$, ..., $(\mathbf{p}^q, \mathbf{t}^q)$, which we call the training set. We wish to train FUZAMP to learn the mapping from \mathbf{p}^k to \mathbf{t}^k , where $k=1,2, \dots, q$. The

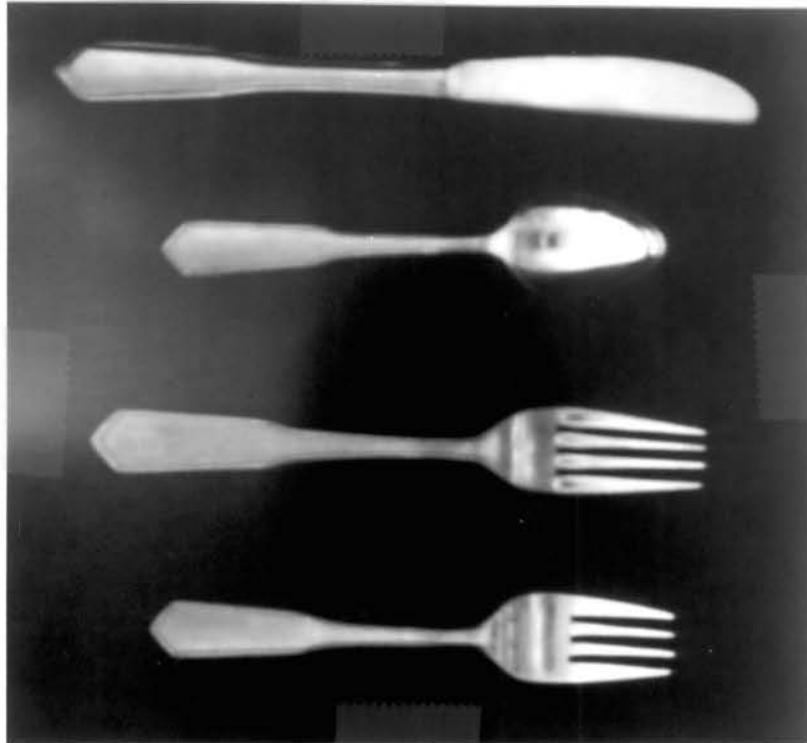


Fig. 5.3 Gray Level Images of Stainless Steel Tableware
(Butter Knife, Tea Spoon, Dinner Fork, Salad Fork)

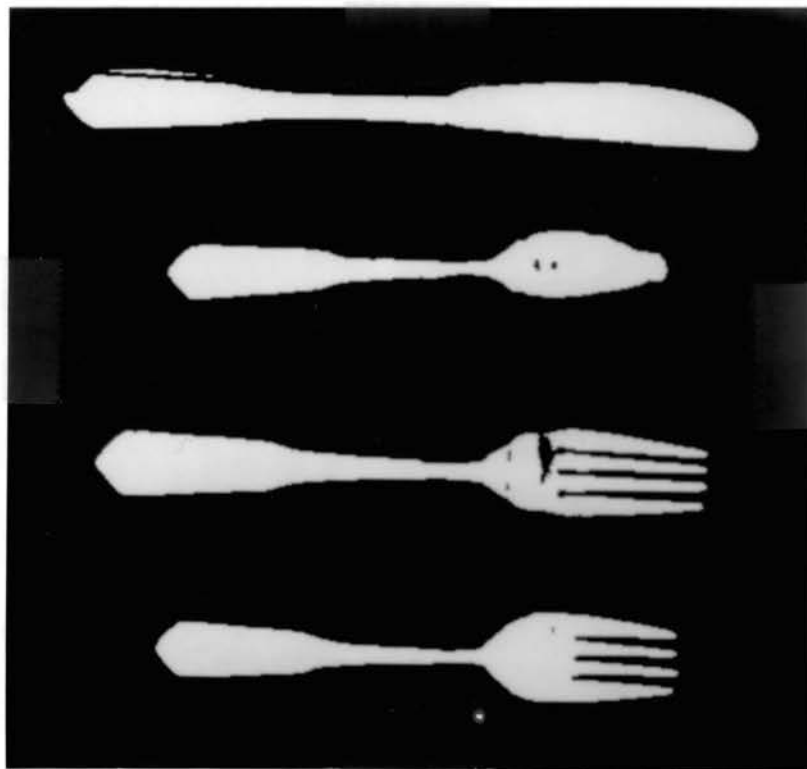


Fig. 5.4 Binary Images of Stainless Steel Tableware
(Butter Knife, Tea Spoon, Dinner Fork, Salad Fork)

training set was presented to FUZAMP such that each fuzzy input vector \mathbf{p} from the training set was presented in the F_0^a field of FUZAMP (Fig. 3.1), while the corresponding fuzzy target vector \mathbf{t} from the training set was presented in the F_0^b field of FUZAMP (Fig. 3.1). We present the training set repeatedly to FUZAMP until the fuzzy weights converge to constants, that is, the bottom-up, top-down, and mapping fuzzy weights in FUZAMP do not change in one training set presentation, called an epoch. After fuzzy weights converge, the desired mappings between inputs and corresponding outputs from the training set will have been established by the FUZAMP network architecture. In our application, the training fuzzy input vector \mathbf{p} has three fuzzy features, namely fuzzy normalized area, fuzzy normalized perimeter, and fuzzy normalized major ellipse radius. Corresponding to these three fuzzy features, the F_0^a field of FUZAMP has three nodes. Without using fuzzy complement coding, the F_1^a field of FUZAMP also has three nodes. The number of nodes in the F_2^a field is initially chosen as one. As fuzzy learning proceeds, the number of nodes in the F_2^a field is automatically increased to establish the mappings between inputs and targets. The F_0^b field of FUZAMP has five nodes, corresponding to five different types of fuzzy objects (silverware pieces) in our application. A similar procedure is used for the comparison neural network, fuzzy ARTMAP (Carpenter and Grossberg, et al., 1991).

Testing of FUZAMP

Once the training process is completed, the performance of FUZAMP is tested. In the testing process we employ a collection of test inputs with known desired targets, called the testing set, which is totally different from the training set. We employ the already trained FUZAMP fuzzy neural network to derive its output in the F_0^b field when test inputs are provided in the F_0^a field. By presenting test inputs from the test set to FUZAMP, we can evaluate the performance of FUZAMP on inputs that have not been seen before by the FUZAMP. A similar procedure is used for the comparison network, fuzzy ARTMAP.

Results

In this section, we present application results using FUZAMP to recognize highly specular silverware. We implemented FUZAMP and the comparison fuzzy ARTMAP on the AdeptOne AGS Vision System on a Motorola 68020 CPU, using the V+ programming language.

In Application 1, FUZAMP was trained with three input features, for the silverware pieces previously described, with three training sets containing, respectively, 25, 50, and 100 exemplars. FUZAMP parameters were set as follows: $\alpha = 0.0001$, $\rho_a = 0.85$, $\rho_b = 1$, $\delta = \gamma = 0.01$, and $\tau \rightarrow \infty$. For comparison, we also employed fuzzy ARTMAP to recognize the same objects, even though fuzzy ARTMAP cannot handle fuzzy data. The training and test inputs to fuzzy ARTMAP were those used for FUZAMP, except that measurement values for each feature were fed to fuzzy ARTMAP without fuzzification. After training, the optimal classification accuracy for fuzzy ARTMAP occurred with $\rho_a = 0.9$, $\rho_b = 1$ and $\alpha = 1$ (Carpenter and Grossberg et al., 1991b). Table 5.1 shows the results of classification accuracy, the number of categories in the F_2^a field, and the CPU time used for training. Classification accuracy was determined by first training the network with 25, 50, or 100 exemplars from a given training set (e.g. training set A) and then presenting a different set of 100 exemplars, called the testing set, and determining the number of correct classifications for that set. The results, both for training and testing, were achieved by presenting randomly the five different types of silverware in the FOV at random locations and orientations, including both right-side-up and right-side-down presentations. Each training set (A, B, C, D, etc.) had different training exemplars. As an example, Table 5.2 shows the training exemplars of data set A and 25 exemplars in the testing set. Note that the testing set exemplars are different from those of the training set. For the same number of training exemplars in Table 5.1, it can be seen that different training sets gave similar performance. For example, with the number of training exemplars equal to 50, FUZAMP typically used six category nodes, required

Table 5.1 Results of Application 1 Using FUZAMP and Fuzzy ARTMAP

No. Training Exemplars		No. F_2^a Categories		Classification Accuracy (%)		CPU Training Time (sec)*	
		FUZAMP	ARTMAP	FUZAMP	ARTMAP	FUZAMP	ARTMAP
25	Training Set A	6	5	98%	93%	8.44	3.32
	Training Set B	5	5	96%	94%	7.85	3.35
	Training Set C	6	5	97%	90%	8.63	3.22
	Training Set D	6	5	98%	90%	8.45	3.53
50	Training Set E	6	5	99%	96%	15.56	4.99
	Training Set F	6	5	98%	90%	14.93	4.91
	Training Set G	5	5	100%	97%	14.19	5.73
	Training Set H	6	5	99%	97%	15.18	5.93
100	Training Set I	8	5	100%	98%	34.27	8.49
	Training Set J	8	5	100%	99%	37.17	11.41

* Adept V+ Program Running on Motorola 68020 CPU

Table 5.2 Training Data Set A and 25 Exemplars in Testing Set
 (Feature 1: normalized area; Feature 2: normalized perimeter;
 Feature 3: normalized major ellipse radius)

Exemplar Index	Training Set			Testing Set		
	Feature 1	Feature 2	Feature 3	Feature 1	Feature 2	Feature 3
1	0.877	0.603	0.696	0.887	0.732	0.719
2	0.884	0.735	0.701	0.604	0.697	0.560
3	0.608	0.656	0.558	0.542	0.504	0.874
4	0.537	0.466	0.883	0.600	0.969	0.893
5	0.604	0.962	0.885	0.732	0.542	0.607
6	0.735	0.560	0.610	0.696	0.459	0.567
7	0.693	0.461	0.566	0.450	0.558	0.601
8	0.436	0.549	0.602	0.967	0.687	0.733
9	0.963	0.715	0.746	0.534	0.702	0.694
10	0.573	0.709	0.693	0.459	0.557	0.448
11	0.759	0.553	0.435	0.559	0.882	0.972
12	0.560	0.879	0.966	0.691	0.870	0.549
13	0.683	0.872	0.547	0.558	0.608	0.460
14	0.702	0.607	0.460	0.875	0.571	0.558
15	0.552	0.566	0.557	0.865	0.595	0.686
16	0.879	0.599	0.683	0.610	0.736	0.557
17	0.867	0.732	0.695	0.549	0.687	0.886
18	0.611	0.687	0.551	0.604	0.453	0.866
19	0.553	0.452	0.875	0.737	0.973	0.616
20	0.601	0.956	0.873	0.692	0.557	0.568
21	0.733	0.561	0.609	0.436	0.468	0.599
22	0.646	0.575	0.544	0.968	0.561	0.728
23	0.476	0.563	0.601	0.542	0.678	0.686
24	0.959	0.695	0.728	0.462	0.551	0.437
25	0.561	0.695	0.698	0.565	0.879	0.967

Indices 1, 6, 11, 16, 21 indicate butter knife
 Indices 2, 7, 12, 17, 22 indicate dinner fork
 Indices 3, 8, 13, 18, 23 indicate salad fork
 Indices 4, 9, 14, 19, 24 indicate soup spoon
 Indices 5, 10, 15, 20, 25 indicate tea spoon

approximately 15 seconds CPU training time, and yielded 99% classification accuracy. It can be seen that the classification accuracy of FUZAMP grew from about 98%, after training on 25 exemplars, to 100% after training on 100 exemplars. The classification accuracy of fuzzy ARTMAP grew from about 92%, after training on 25 exemplars, to about 97% after training on 100 exemplars. FUZAMP achieved a higher test set accuracy using only a small number of training exemplars. In contrast, fuzzy ARTMAP needed more training exemplars to reach the comparable accuracy. On the other hand fuzzy ARTMAP required slightly fewer category nodes and considerably less CPU training time to converge than did FUZAMP.

In Application 2, FUZAMP was trained using only two features of silverware images, namely area and perimeter. Its parameters were set as follows: $\alpha=0.0001$, $\rho_a=0.85$, $\rho_b=1$, $\delta=\gamma=0.01$, and $\tau \rightarrow \infty$. We also trained the fuzzy ARTMAP to recognize these objects using only area and perimeter features. The optimal classification accuracy for fuzzy ARTMAP occurred with $\rho_a=0.9$, $\rho_b=1$, and $\alpha = 1$. Table 5.3 shows the results. Using only two features, FUZAMP produced very good classification accuracy as the number of training exemplars increased. In comparison, fuzzy ARTMAP did not achieve as good accuracy, although CPU training time was less.

In Application 3, the 25 training exemplars of training set D in Table 5.1 were used for training FUZAMP with different spread values for the triangular membership function and employing three features. FUZAMP was run on the training data set with three features. Table 5.4 demonstrates the effect of the spread values. We note that excessive spread results in low classification accuracy, with spread values of 0.01 yielding the highest accuracy. Since spread values reflect uncertainty existing in given measurements, large spread values should be chosen if the machine vision measurements yield large uncertainty, and vice versa. Note that since each machine vision measurement was fuzzified by a range of spread values, most spread values were incommensurable with the overall variations of each feature.

Table 5.3 Results of Application 2 Using FUZAMP and Fuzzy ARTMAP

No. Training Exemplars		No. F_2^a Categories		Classification Accuracy (%)		CPU Training Time (sec)*	
		FUZAMP	ARTMAP	FUZAMP	ARTMAP	FUZAMP	ARTMAP
25	Training Set A	5	5	90%	85%	5.22	2.60
	Training Set B	5	5	92%	83%	5.64	2.53
	Training Set C	6	5	92%	85%	6.38	2.64
	Training Set D	6	5	91%	86%	5.76	2.59
50	Training Set E	6	5	96%	90%	11.35	3.84
	Training Set F	5	5	97%	91%	10.92	3.76
	Training Set G	6	5	96%	92%	11.80	3.70
	Training Set H	6	5	96%	89%	10.39	3.89
100	Training Set I	7	5	98%	94%	28.10	6.28
	Training Set J	7	5	99%	94%	30.52	7.54

* Adept V+ Program Running on Motorola 68020 CPU

Table 5.4 Results of Application 3 for FUZAMP

Spread Values	No. Categories	% Test Set Accuracy
0.1	6	70%
0.05	6	93%
0.01	6	98%
0.005	6	94%
0.001	6	91%
0	5	90%

Dish Cleanliness Inspection Using FUZAMP

In our dish cleanliness inspection problem, sample images used to train and test FUZAMP were obtained from the same vision system as for our silverware recognition problem. We used five different rectangularly shaped ceramic and plastic dishes photographed in Fig. 5.5. These dishes were large plastic spacers, large ceramic dishes, small plastic dishes, small ceramic dishes, and small plastic spacers. Plan view dimensions of each dish type are 192mm×128mm for large plastic spacers and large ceramic dishes, respectively, 123mm×92mm for small plastic dishes, 128mm×98mm for small ceramic dishes, and 125mm×95mm for small plastic spacers. In order to investigate the performance of FUZAMP, we set a rectangle inspection zone located around the center of each type of dish, with dimension 65mm×45mm for small dishes, and 120mm×65mm for large dishes, respectively. The purpose of the inspection zone was to avoid the shadows caused by the turned-up edges of the dishes. The images of these inspection zones were obtained based on the following sequential procedures: 1. start the system and let it warm-up for half an hour; 2. employ fuzzy logic controllers to set the vision parameters "gain", "offset", and "threshold", as described in Chapter IV; 3. for a given dish type, present a series of dishes, each with various real food particles, in the center of the camera FOV, with each dish at approximately the same location and orientation. Repeat for every dish type; 4. employ the AGS machine vision image processing software to obtain the gray level histogram for each individual dish image, and store the histogram in memory. The image of each dish was processed in the same way, namely:

1. Obtain a gray level histogram for the entire inspection zone of a dish.
2. Find the two gray levels, corresponding to the positive and negative histogram gradient directions, with the absolute value of the ratio of the change in pixel frequency to the change in gray level equal to 15, as illustrated in Fig. 5.6.



Fig. 5.5 Five Different Rectangularly Shaped Ceramic and Plastic Dishes
Top to Bottom, Left to Right: Small Plastic Dish, Small Ceramic Dish, Small Plastic
Spacer, Large Ceramic Dish, Large Plastic Spacer

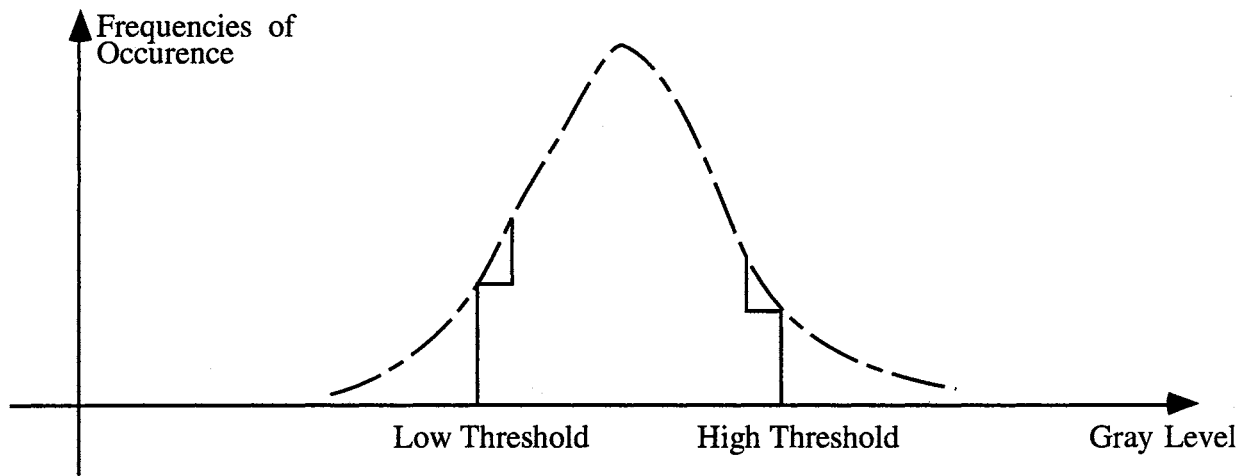


Fig. 5.6 Gray Level Histogram of A Dish

Higher gray level values correspond to whiter images and lower gray level values correspond to blacker images.

3. Examine the dish gray levels to see whether they are all in the expected range for a clean dish (found experimentally for each type of dishes). If these measured gray levels are out of this range, the dish is classified as dirty.
4. Otherwise, using thresholds in Step 2, count the pixels with gray levels above the high threshold, below the low threshold, and between the high and low thresholds, respectively.
5. Employ the three pixel counts of Step 4 as three features, respectively. Each feature is normalized by dividing its pixel count by a maximum possible pixel count. The normalized features are then fuzzified by appropriate choices of the triangular membership functions, such that FUZAMP produces an optimal inspection accuracy.
6. To improve the sensitivity of the inspection system to various dirty food particles, classify the dishes using FUZAMP as (i) clean, (ii) dirty with high gray level food particles, (iii) dirty with low gray level food particles, (iv) dirty with medium gray level food particles, or (v) dirty with both high and low gray level food particles. Using these five categories, instead of just two (i.e., "clean" or "dirty") allows the system more sensitivity in selecting "clean" dishes. After the system is suitably trained by employing these five categories, we are able to achieve good results by further classifying as dirty dishes falling into categories (ii) to (v), and dishes falling into category (i) as "clean".

Sample images of different types of dishes with various real food particles, including some very small food particles of several pixels in size were manually grouped into the five classes (i-v) described above. FUZAMP was trained with three numbers of training exemplars, 15, 25, and 35, for each type of dish. We employed 100 testing exemplars, different from those in the training sets, to test the performance of FUZAMP. Optimal

FUZAMP parameters was set as follows: $\alpha = 0.0001$, $\rho_a = 0.9$, $\rho_b=1$, $\delta = \gamma = 0.02$, and $\tau \rightarrow \infty$. For comparison, we also employed fuzzy ARTMAP to inspect the dish cleanliness. The training and testing inputs to fuzzy ARTMAP were those used for FUZAMP, except that measurement values for each feature were fed to fuzzy ARTMAP without fuzzification. After training, the optimal inspection accuracy for fuzzy ARTMAP occurred with $\rho_a=0.95$, $\rho_b=1$, and $\alpha=1$. Table 5.5 shows the results of inspection accuracy. Inspection accuracy was determined by manually classifying a dish as "clean" or "dirty" and then comparing FUZAMP's results. Using a small number of training exemplars, FUZAMP achieved higher inspection accuracy than fuzzy ARTMAP. As the number of training exemplars increased, inspection accuracy achieved by fuzzy ARTMAP was comparable to that achieved by FUZAMP. Note that inspection accuracy achieved by both FUZAMP and fuzzy ARTMAP was affected by the system resolution, especially for the situations where food particles were very small, or food particles and dish surfaces had similar gray level.

Summary

In this chapter, the new fuzzy neural network, FUZAMP, was applied to recognize fuzzy objects and to inspect dish cleanliness using machine vision. FUZAMP performed well when used to handle uncertainty caused by variations in imaging by the machine vision system. The performance of FUZAMP, a true fuzzy neural network, was compared to that of fuzzy ARTMAP, which is not truly fuzzy. The recognition and inspection accuracy of FUZAMP was found to be higher than that of fuzzy ARTMAP, while the CPU training time for fuzzy ARTMAP was significantly less than that for FUZAMP. This is because for a truly fuzzy neural network such as FUZAMP, considerably more mathematical operations are required.

Table 5.5 Inspection Accuracy (%) for Dish Cleanliness Inspection
Using FUZAMP and Fuzzy ARTMAP

Dish Type	Number Training Exemplars					
	15		25		35	
	FUZAMP	ARTMAP	FUZAMP	ARTMAP	FUZAMP	ARTMAP
Small Plastic Spacer	87%	82%	91%	90%	92%	92%
Small Ceramic Dish	80%	76%	87%	85%	87%	86%
Small Plastic Dish	81%	73%	86%	85%	86%	86%
Large Plastic Spacer	88%	83%	92%	89%	93%	92%
Large Ceramic Dish	83%	78%	88%	86%	88%	88%

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

The main goal of this work was to develop new unsupervised and supervised fuzzy neural networks for fuzzy patterns. Investigation of new fuzzy neural network architecture concentrated on extending Adaptive Resonance Theory (ART) networks because of the attractive performance of ART. The new unsupervised and supervised fuzzy neural networks were evaluated by simulations and real machine vision applications. Following are the major contributions of this research and some recommendations.

Conclusions

1. An unsupervised fuzzy neural network for fuzzy patterns, termed FUZART, has been proposed. FUZART employs the two stages of self-organization and decision making. It accepts fuzzy input patterns, numerical input patterns, or both simultaneously, and provides output decisions in terms of membership values. As an extension of fuzzy ART, concepts of fuzzy norm, fuzzy subset function, fuzzy resonance, and fuzzy complement coding are presented and defined for FUZART. FUZART has the ability to learn on-line using only a few training epochs. Simulation results demonstrate the ability of FUZART to provide reasonable clustering decisions for fuzzy patterns.
2. A new supervised fuzzy neural network scheme, called FUZAMP, has been developed to handle either fuzzy input patterns, numerical input patterns, or both simultaneously using a single system. FUZAMP can realize fast and efficient training for fuzzy data and possesses to a high degree all the desired properties of a

supervised fuzzy neural networks. It can be employed as a fuzzy inference engine and can be applied in fuzzy pattern recognition problems using linguistic knowledge described by fuzzy rules and numerical data sampled by measurement instruments.

3. A multi-layer MIMO fuzzy logic controller has been proposed and implemented to realize automatic adjustment of the camera parameters "gain" and "offset" to compensate for power fluctuation, changes in ambient light, and camera sensitivity drift in our machine vision system. The multilayer FLC yields faster response with less overshoot than that of a conventional single layer FLC, and provides excellent camera performance.
4. FUZAMP has been used to deal with situations where the available training data from a machine vision system includes uncertainty. It performs well when used to recognize different types of fuzzy objects presented at different locations and orientations in the camera Field of View. In addition, FUZAMP has been implemented to correlate human evaluations with machine evaluations of the cleanliness of dishes. Results are compared to those obtained using the so-called fuzzy ARTMAP neural network, with FUZAMP achieving better accuracy than the fuzzy ARTMAP using the same training exemplars.

Recommendations

While this research focused on development of new unsupervised and supervised fuzzy neural networks, with applications in machine vision recognition and inspection problems, applications of fuzzy neural networks to practical problems can be extended to fuzzy expert systems, fuzzy logic control, and fuzzy classification. Integration of fuzzy sets, neural networks, and genetic algorithms for intelligent systems may be a good research area for future investigation. While speech recognition is a very complex problem, it is possible that FUZAMP may prove to be a significant advance in solving this

problem. Moreover, FUZAMP and FUZART may hold significant promise in advancing the state-of-the-art in intelligent control systems.

REFERENCES

1. Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., and Rosen, D. B. (1991a). FUZZY ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. *IEEE Transactions on Neural Networks*, **3**, 698-713.
2. Carpenter, G. A., Grossberg, S., and Rosen, D. B. (1991b). FUZZY ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. *Neural Networks*, **4**, 759-771.
3. Carpenter, G. A. and Grossberg, S. (1987). A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computer Vision, Graphics, and Image Processing*, **37**, 54-115.
4. Chen, B., and Hoberock, L., L. (1996a). Machine Vision Recognition of Fuzzy Objects Using a New Fuzzy Neural Network. Accepted for Presentation at *the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 22-28.
5. Chen, B., and Hoberock, L., L. (1996b). A Fuzzy Neural Network Architecture for Fuzzy Control and Classification. Submitted for Presentation at *the 1996 International Conference on Neural Networks*, Sheraton, Washington, DC, June 2-6, 1996
6. Chen, B., and Hoberock, L., L. (1995). Fuzzy Logic Controller for Automatic Vision Parameter Adjustment in a Robotic Dish Handling System. In *Proc. the 10th IEEE International Symposium on Intelligent Control*, Monterey, CA, Aug. 27-29.
7. Chiu, S., and Chand, S. (1993). Adaptive Traffic Signal Control Using Fuzzy Logic, In *Proc. FUZZ-IEEE'93*, 1371-1376.

8. Feng, K., and Hoberock, L., L. (1992). A Modified ARTMAP Network, With Applications to Scheduling of a Robot-Vision Tracking Systems. In *Proc. 1992 ASME Winter Annual Meeting*, Anaheim, CA, Nov. 8-13. To Appear in *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*
9. Hayashi, Y., Buckley, J. J., and Czogala E.(1993). Fuzzy Neural Network with Fuzzy Signals and Weights. *International J. of Intelligent Systems.*, **8**, 527-537.
10. Hellendoorn, H., Thomas, C. (1993). Defuzzification in Fuzzy Controllers. *Journal of Intelligent and Fuzzy Systems*, **1**, 109-123.
11. Ishibuchi, H., Fujioka, R., and Tanaka, H. (1993). Neural Networks That Learn from Fuzzy If-Then Rules. *IEEE Transactions on Fuzzy Systems*, **1**, 85-97.
12. Ishibuchi, H., Tanaka, H., and Okada, H. (1994a). Interpolation of Fuzzy If-then Rules by Neural Networks, *International J. of Approximate Reasoning*, **10**, 3-27
13. Ishibuchi, H., Morioka, K., and Tanaka, H. (1994b). A Fuzzy Neural Network with Trapezoid Fuzzy Weights. In *Proc. FUZZ-IEEE'94* (Orlando, Florida), pp.228-233, June 26-29.
14. Keller, J. M., and Tahani, H. (1992a). Implementation of Conjunctive and Disjunctive Fuzzy Logic Rules with Neural Networks. *International J. of Approximate Reasoning*, **6**, 221-240.
15. Keller, J. M., and Tahani, H. (1992b). Back Propagation Neural Networks for Fuzzy Logic, *Information Science*, **62**, 205-221.
16. Kim, Y. S. and Mitza S. (1993). Integrated Adaptive Fuzzy Clustering Algorithm. In *Proc. FUZZ-IEEE'93*, 1264-1268.
17. Kosko, B. (1986). Fuzzy Entropy and Conditioning. *Information Science*, **29**, 165-174.
18. Lee, C. C. (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller -Part I and Part II. *IEEE Transactions on Systems, Man, and Cybernetics*, **20**, 404-435.

19. Mitra, S., and Pal, S. K. (1994). Self-Organizing Neural Network as a Fuzzy Classifier. *IEEE Transactions on Systems, Man, and Cybernetics*, **24**, 385-399.
20. Pedrycz, W. (1991). Fuzzy Logic in Development of Fundamentals of Pattern Recognition. *International Journal of Approximate Reasoning*, **5**, 252-264.
21. Pham, D. T., and Bayro-Corrochano, E. J. (1994). Self-Organizing Neural-Network-Based Pattern Clustering Method with Fuzzy Outputs. *Pattern Recognition*, **27**, 1103-1110.
22. Rousselot, J. P., Balmat, J. F., and Gut, A. (1993). A Fuzzy Logic Knowledge-Based System in Naval Decision-Support Aids. In *Proc. FUZZ-IEEE'93*, 833-838.
23. Sahoo, P. K., Soltani, S., Wong, A. K. C., and Chen, Y. C. (1988). A Survey of Thresholding Techniques. *Computer Vision Graphics Image Processing*, **41**, 233-260.
24. Simpson, P. K. (1993). Fuzzy Min-Max Neural Networks - Part 2: Clustering. *IEEE Transactions on Fuzzy Systems*, **1**, 32-45.
25. Umamo, M., and Ezawa, Y. (1992). Execution of Approximate Reasoning by Neural Network. *International J. of Approximate Reasoning*, **6**, 163-178.
26. Wang, L. X. (1994). *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice-Hall, Inc.
27. Yasunobu, S., and Miyamoto, S. (1985). Automatic Train Operation System by Predictive Fuzzy Control. *Industrial Applications of Fuzzy Control* (Michio Sugeno), Ed, Amsterdam: North-Holland, 1-18.
28. Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, **8**, 338-353.
29. Zimmermann, H. (1991). *Fuzzy Set Theory and its Applications*. Norwell, MA: Kluwer Academic Publishers.

APPENDIX

BRIEF GLOSSARY FOR FUZZY LOGIC

(Wang, 1994; Zimmermann, 1991)

Fuzzy Set: Let U be the universe of discourse. A fuzzy set F in U is characterized by a membership function $\mu_F : U \rightarrow [0,1]$, with $\mu_F(u)$ representing the degree of membership of $u \in U$ in the fuzzy set F . A fuzzy set may be viewed as a generalization of the concept of an ordinary set (that is, a crisp set) whose membership function assumes only two values, namely 0 and 1, $\{0, 1\}$.

Fuzzy Number: The term fuzzy number is used for any fuzzy set F with the following specific properties:

1. There exists exactly one $u_0 \in U$ such that $\mu_F(u_0) = 1$
2. For any real variable $u, v, u \leq v$, and any $w \in [u, v]$, $\mu_F(w) \geq \min(\mu_F(u), \mu_F(v))$

Fuzzy Vector: A vector whose components are fuzzy numbers is called a fuzzy vector.

Support, Mean, and Fuzzy Singleton: The support of a fuzzy set F with membership function $\mu_F(u)$ is the crisp set of all points $u \in U$ such that $\mu_F(u) > 0$. The mean of a fuzzy set F is the point(s) $u \in U$ at which $\mu_F(u)$ achieves its maximum value. If the support of a fuzzy set F is a single point in U at which $\mu_F = 1$, F is called a fuzzy singleton.

The Extension Principle: Let U and V be two universes of discourse and f be a mapping from U to V . For a fuzzy set A in U , the extension principle defines a fuzzy set B in V by $\mu_B(v) = \sup_{u \in f^{-1}(v)} [\mu_A(u)]$. That is, $\mu_B(v)$ is the superior of $\mu_A(u)$ for all $u \in U$ such that $f(u) = v$, where $v \in V$ and we assume that $f^{-1}(v)$ is not empty. If $f^{-1}(v)$ is empty for some $v \in V$, define $\mu_B(v) = 0$.

Linguistic Variables: If a variable can take words in natural languages (for example, small, fast, and so on) as its values, this variable is defined as a linguistic variable. These

words are usually labels of fuzzy sets. A linguistic variable can take either words or numbers for its "values".

Fuzzy Rule Base: A fuzzy rule base consists of a collection of fuzzy IF-THEN rules in the following form:

$$R^{(L)} : \text{IF } x_1 \text{ is } F_1^L \text{ and } \cdots \text{ and } x_n \text{ is } F_n^L, \text{ THEN } y \text{ is } G^L$$

where F_i^L and G^L are fuzzy sets in the universe of discourse U_i and V , respectively, and $(x_1, \dots, x_n)^T \in U_1 \times \cdots \times U_n$ and $y \in V$ are linguistic variables.

Fuzzy Inference Engine: In a fuzzy inference engine, fuzzy logic principles are used to combine the fuzzy IF-THEN rules in the fuzzy rule base into a mapping from fuzzy sets in $U = U_1 \times \cdots \times U_n$ to fuzzy sets in V .

Fuzzification: Fuzzification performs a mapping from a crisp point $(x_1, \dots, x_n)^T \in U$ into a fuzzy set A in U .

Defuzzification: Defuzzification performs a mapping from fuzzy sets in V to a crisp point $y \in V$.

VITA 

Baoshan Chen

Candidate for the Degree of

Doctor of Philosophy

Thesis: UNSUPERVISED AND SUPERVISED FUZZY NEURAL NETWORK ARCHITECTURE, WITH APPLICATIONS IN MACHINE VISION FUZZY OBJECT RECOGNITION AND INSPECTION

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Fujian, China, March 28, 1963, the son of Xiuqin Li and Shilin Chen.

Education: Graduated from Huian First High School, Fujian, China, in July, 1979; received Bachelor of Science Degree in Mechanical Engineering from Zhejiang University in July, 1983; received Master of Science Degree in Mechanical Engineering from Huaqiao University in May, 1986; completed requirements for the Doctor of Philosophy Degree at Oklahoma State University in May, 1996.

Professional Experience: Graduate Research Associate, School of Mechanical and Aerospace Engineering, Oklahoma State University, from August, 1993, to January, 1996; Assistant Professor and Senior Research Engineer, Department of Precision Mechanical Engineering, Huaqiao University, from July, 1986, to August, 1993.